سلسلة تعليمية

من الصفر حتى الاحتراف



Microsoft

Visual Basic.Net

فيجوال بيسك . نت

للمبتدئين

اعداد

محمد حازم الشميري

www.learn-barmaga.com

سلسلة تعليمية

من الصفر حتى الاحتراف

Visual Basic.Net

فيجوال بيسك . نت

للمبتدئيين

الطبعة الاولى

١٤٣٨هـ – ٢٠١٦م

اعداد / محمد حازم الشميري

نبــذة عــن المؤلــف



محمد حازم الشميري

خريج كلية العلوم الادارية / بك محاسبة – جامعة تعز – ٢٠١٤ م

الجمهورية اليمنية – تعز

الايميل : profionaldhim@gmail.com

موبايل : ٠٠٩٦٧٧١٦٨٧٥٦٨١ - وتس اب : 00967733014747

فيسبوك : www.fb.com/profionaldhim

تويتر : www.twitter.com/profionaldhim1

الاهداء

اهدي هذا الكتاب والمشروع المتواضع الى من كان لي كنز العلم والمعرفـــة الـــى تـــاج راسي امي الغالية يامن تستحقين كل هذا الشكر والتقدير الى ابي الغالي جيفـــارا حياتـــي العلمية من كان له الدور العظيم في ايصالي لهذا المستوى العلمـــي الـــى مكتبـــة وســـر ثقافتي وتشجيعي العلمي وسبب حبي وهوايتي لهذا المجال اخـــي الغالـــي المهنـــدس/ عبدالبـــاسط حازم الشميـــري ولكل اصدقائـــي وزملائـــي مـــن زرعـــوا فينـــي الثقـــة العاليـــة ولكـــل مـــن تعـــاون معـــي بانجـــاز هـــذا المشـــروع بـــدون صفـــة أو استثنـــاء اشكركم جميعاً احبتـــي ....

محبكـــم

م / محمـــد حـــازم الشميـــري

التقـديـم :

الحمد لله الذي بحمده تتم النعم وننال بفضله الأمنيات ، لقد أسعدني وغُمر قلبي بالسرور عندما علمتُ بخبر إصدار هذا الكتاب للأخ المبرمج محمد الشميري والذي سيتحدث عن لغة البرمجة Visual Basic.Net والتي تعتبر إحدى اللغات ضمن عائلة الـ NET. وهي من أهم لغات البرمجة الشهيرة تطبيقاً في الجانب العملي ، ومماأسعدني أكثر هو أن يعطيكم الكاتب بعضاً من خلاصة خبرته وبحثه في مجال البرمجة وبالتحديد البرمجة بإستخدام لغة Visual Basic.Net ، حيث أنه يحتفظ في جعبته بالكثير من المعرفة من خلال خبرته في مجال تصميم العديد من الأنظمة المحاسبية الناجحة ولديه أيضاً إسهامات علمية كثيرة ومتميزة كالمقالات العلمية وإنشاء مشاريع في مجالات مختلفة ، لن أتحدث عن الكاتب كثيراً لأنه سيقدم نفسه لكم من خلال محتوى هذا الكتاب ...

ومايزيد الأخ محمد الشميري تألقاً وتميزاً هو أنه يمتلك أسلوباً فريداً ومقنعاً في طرح المواضيع وبشكل سلس وبسيط ، بحيث يُمكن القارئ له من إستيعاب ومتابعة شرح المواضيع مهما كان مستواه التعليمي ...

أنا كمهندس في مجال تكنولوجيا المعلومات ومتخصص في تصميم الأنظمة والمواقع الإلكترونية ، أرى أن هذا الكتاب سيكون إضافة نوعية في هذا الحقل التعليمي مما سيرفده أكثر وأكثر ، أيضا سيساهم بالإرتقاء بفكرة المحتوى العربي الرقمي وذلك لصعوبة الحصول على المراجع والمصادر التي يُراد الحصول على المعلومات منها وعلى وجه الخصوص في لغات البرمجة والتي صارت إحدى لغات التخاطب مع الجانب الرقمي من العالم ومصدراً لإنعاش ثورة نظم المعلومات ...

يتميز هذا الكتاب عن غيره بأسلوب الكاتب في طرح المعلومات وتسلسل الأفكار ، حيث أنه يعتبر دورة متكاملة لتعلم وإحتراف لغة البرمجة Visual Basic.Net وستبدأ معه كمبتدئ ليأخذك خطوة بخطوة لكي ترتقي معه وتصل إلى مرحلة الإحتراف وإلى تصميم برامجك الخاصة ...

وأختتم كلامي هنا بتقديم خالص تمنياتي لكم أعزائي القراء بالتوفيق وأن تحصدوا ذلك الكم الهائل من المعلومات من خلال هذا الكتاب ، وأقدم جزيل شكري وحبي لأخي محمد الشميري مؤلف هذا العمل على ماقدمه لنا بين أغلفة هذا الكتاب وأدعوه بأسمي وبالنيابة عن الكثيرين بمواصلة تقديم المزيد للإفادة ، كما أود تهنئته على هذا العمل الجبار وأتمنى له المزيد من التألق والنجاح ...

خالص تحياتي ..

✍ بقلم م. أشرف وهيب العبسي

مهندس تكنولوجيا المعلومات

مقدمة

"سبحانك لاعلم لنا الا ماعلمتنا ، انك انت العليم الحكيم " ... وبعد

عزيزي القارئ ، ها انا اعــددت لكــم هذا الكتاب المتواضــع في اصداره الاول لي ، والذي يتناول فـــي محتـــواه دروس تعليميـــة عـــن لغـــة البرمجـة Visual Basic.Net التـــي لطالمـــــا احببتهـــا منـــذ أن عرفتهـــا منــذ مرحلتـــي الدراسيـــة فـــي الصفـــوف الاولـــى مــن الثانـــوية العــامة وبداءت اجيـــد واحتـــرف التعـامل معهـــا ، لغة Visual Basic.Net لهـا مميزاتهـــا الخاصـــــــة ولهـا عيوبهـــا كـــذلك ، فـــان لغـــة Visual Basic.Net بجميـــع اصداراتهـــا 2005 ,2008,2010,2013 جميعها تؤدي نفس المهمــة ولايوجد فيمـا بينها الا فــــروق بسيطـــة ويتضمن مشروع الكتـاب الـذي هو بين يديكم دروس للمبتدئيين فـــي لغـــة Visual Basic.Net سيتنـــاول هذا الكتاب العديـد من الفصـــول النظرية والعملية التي تهم القـارئ المبتدئ و سيتم تخصيص الفصل الاول لكل ماهو ماهو يتعلق باساسيات لغـــة كما انه سيتم تخصيص الفصل الثاني للغة البرمجـــة وبالفصل الثالث وسنتحدث عن التعامل مع صندوق الادوات Tool Box وسنتحدث عــن الخصائـــص والاحداث المتعلقة بـ Tool Box وبالفصل الرابع سنتحدث عن التعامـــل مع النمـــاذج والخصائــــص والطرق والاحداث كذلك سنتحدث عن نماذج Midi Parent والثيمـــات وسيتعلق بذلك من وحــــدات برمجية كذلك التعرف على تطبيقات console application وكــــذلك المتغيــــرات والثوابت وكـذلك المصفـــوفات وكذلك التفرع والتكرار و مقدمة عن تصحيح الاخطاء ، بالقراءة المكتسبة نصنع المستحيل بحب القراءة نـبدع ونصنع المستحيل لم يكن هذا المجال تخصصي بقدر ماكان صديقي ورفيق حيـاتي وحلمـــي المفقـــــــود نسال الله أن يكـــون هذا الكتاب مفيداً فما كان فيه من صواب فمن الله وما كان مــن خطــــأ فمن نفســــي والشيطــــان والله الهادي الى سواء السبيل .

نلقاكم أن شاء الله في اصدار اخر للبرمجة المتوسطة في لغة Visual Basic.Net.

وبالاخير اتمنـــا أن ينـــال هذا الكتاب اعجابكم

" واخر دعوانا أن الحمدلله رب العالمين "

محمد حازم الشميري

المحتويات



تعرف على Visual Basic.Net

هذا الفصل كما سترى عزيزي القارئ فصل يتضمن المواضيع الاساسية التي هي عبارة عن جزء من لغة البرمجة Visual Basic.Net والذي ستتعرف من خلاله على :

- **محتويات إطار العمل NET Framework**
- **بيئة التطوير Visual Basic.Net**
- **كتابة برنامجك الأول**

محتويات اطار العمل Net Framework.

تقدم بيئة **Net**. ثلاثة عناصر لها يمكن أن تتحكم في دورة حياة البرنامج المعتمد على بيئة **Net**. وهذه التقنيات الثلاث يرمز لها بالرموز الاتية اختصارا لها :

- **CLR, CTS and CLS.**

CLR	اختصار لـ **Common Language Runtime**
CTS	اختصار لـ **Common Type System**
CLS	اختصار لـ **Common Language Specification**

CLR: حيث تعتبر مهمتها الاساسية في الانواع والمكونات الاساسية لبيئة **NET**. ك المكتبات والدوال وحجز المتغيرات في الذاكرة وتقسيم الذاكرة والتعامل معها والتنفيذ ال **Threads** اضافة لبعض اختصارات الامان فتعتبر **CLR** هي المرحلة الاساسية لترجمة اوامر **Net**. وتنفيذها على جهاز الكمبيوتر.

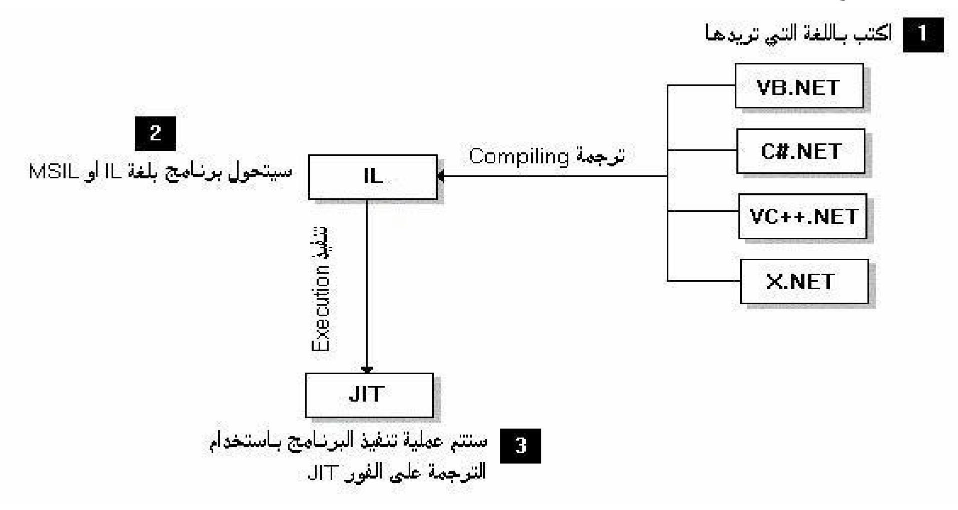
CTS: وتعد ال **CTS** مختصة بالتعامل مع انواع البيانات المدعومة للعمل ضمن بيئة **Net**.

CLS: فهي مختصة للتعامل فقط مع البيانات القياسية والمدعومة من جميع اصدارات بيئة **Net**.

وبهذا نستطيع ان نقول ان البرنامج المصمم باستخدام **Visual Basic.Net** مثلاً ويستخدم فقط الأنواع المتاحة في CLS هو متوافق تماماً للعمل على نفس توزيعة Net. من اجل العمل على Linux المسماه mono اما لو خرجت خارج نطاق CLS الى CTS فلن تضمن ان يتم تنفيذها بنفس الكفاءة في كل الأنظمة المختلفة.

الترجمة على الفور JIT

طبعا الترجمة الفورية تعتبر نقلة نوعية في عالم البرمجة وهي تقنية ، **Just In Time Compiling – (JIT)** هو أسلوب الترجمة على الفور **NET**. يقوم بترجمة البرنامج عند تنفيذه حيث ينتج أفضل شيفرة تتناسب مع الجهاز الذي سيعمل عليه البرنامج مما ينتج عنه نتائج إيجابية جيدة جدا (هذا عند الحديث عن تحسين الكفاءة **Optimization**)
وحتى تعلم كيف يحدث ذلك تابع (الشكل ١-١) "



شكل (١-١) مراحل ترجمة وتنفيذ البرنامج .

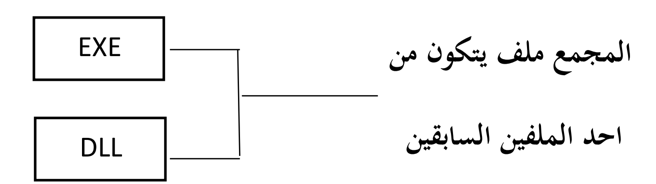
تتم عملية الترجمة من خلال كتابة الكود البرمجي باي لغة برمجية تريدها ولتكن على سبيل المثال **Visual Basic.Net**
vc#.Net, ...الخ

فانك سوف تلاحظ أن تلك العملية مرت بثلاث مراحل وهي :

- كتابة الشفرة (**code**) .
- ترجمة **Compiling** ومنها ستتحول الترجمة الى برنامج بلغة **IL** أو **MSIL**
- التنفيذ وهنا سيتم تنفيذ البرنامج باستخدام الترجمة **JIT**

طبعاً انت عندما تقوم بكتابة الشفرة وتاتي بعدها عملية الترجمة التي تقوم بتحويل ملف البرنامج الى صيغة اشبه بكثير من الملف التنفيذي Executable File مكتوب بلغة تسمى **Microsoft Intermediate Language** (تختصر ب **IL** أو **MSIL**) بهذه الحالة هي تحتوي على شفرات البرنامج ولكنها غير قابلة للتنفيذ الا بوجود مترجم على الفور **JIT Compiler** حينها سيقوم **JIT** بترجمة هذا الملف الثنائي الى لغة الالة (الصفر والواحد) حيث يتمكن من خلال الحصول على افضل المعطيات التي تتناسب مع جهازك الحالي .

المجمعات Assemblies



بحيث أن هذا الملف يحتوي على شفرة البرنامج بعد الترجمة ومارافقها من ملفات مصادر Resource Files وصور ورسومات وصفحات HTML .....الخ وكل هذه العناصر بامكانك أن تضعها في ملف واحد .

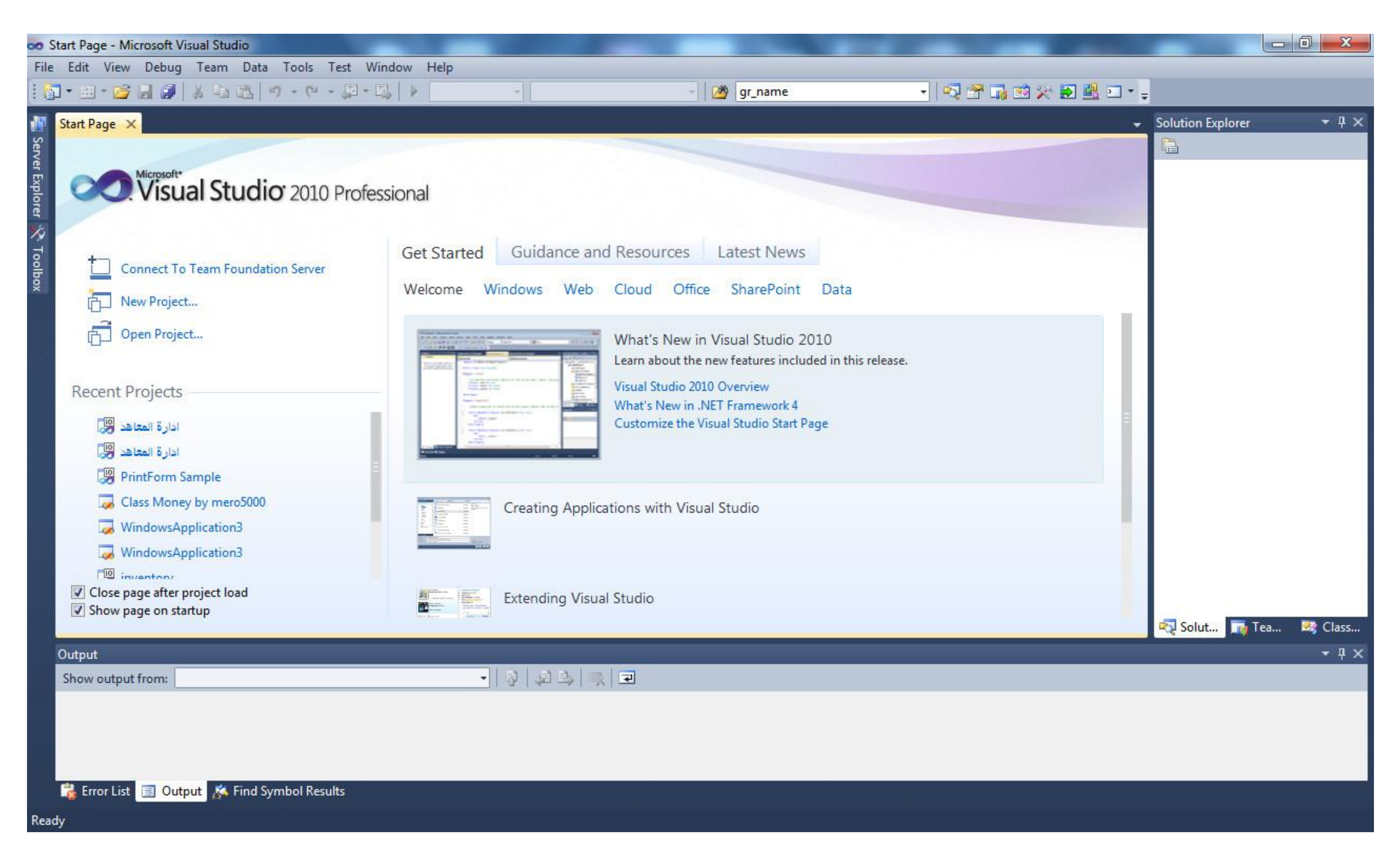
كل مجمع يحتوي على برنامج واحد وبامكانك أن تقوم بعملية اضافة مجمعات اخرى بمعنى انه يمكن لبرنامجك أن يدمج في داخله برنامج اخر .

بيئة التطوير Visual Studio .NET

طبعا بالنسبة لبيئة التطوير في لغة Visual Basic.Net تحتوي على تفاصيل كثيرة عزيزي القارئ والتي تقوم بتعريف الشخص بكافة نوافذ لغة Visual Basic.Net سنتطرق في هذا الفصل بشكل سريع للعديد من النوافذ في بيئة التطوير وسنترك مجالا للقارئ لكي يواصل عملية قراءة مكتسبة ونحن نعلم أن باستطاعته ذلك ، طبعا من المعلوم أن معظم لغات برمجة Net. بما فيها لغة (Visual Basic.Net, VC#.Net ....الخ جميعها تعمل على بيئة تطوير متكاملة ورائعة جدا من الخدمات و الادوات التي تسهل لك حياتك البرمجية ، طبعاً بيئة التطوير في لغة Visual Basic.Net واسعة جدا وتحتاج الى شرح مكثف أن لم يخصص لها كتاب متكامل وذلك لكبر حجمها اما ماسنتناوله نحن هنا فهو ماسيتعلق بلغة البرمجة Visual Basic .NET فقط ولن نذكر الا الادوات والنوافذ التي سنستخدمها نحن هنا وباختصار شديد .

نوافذ بيئة التطوير

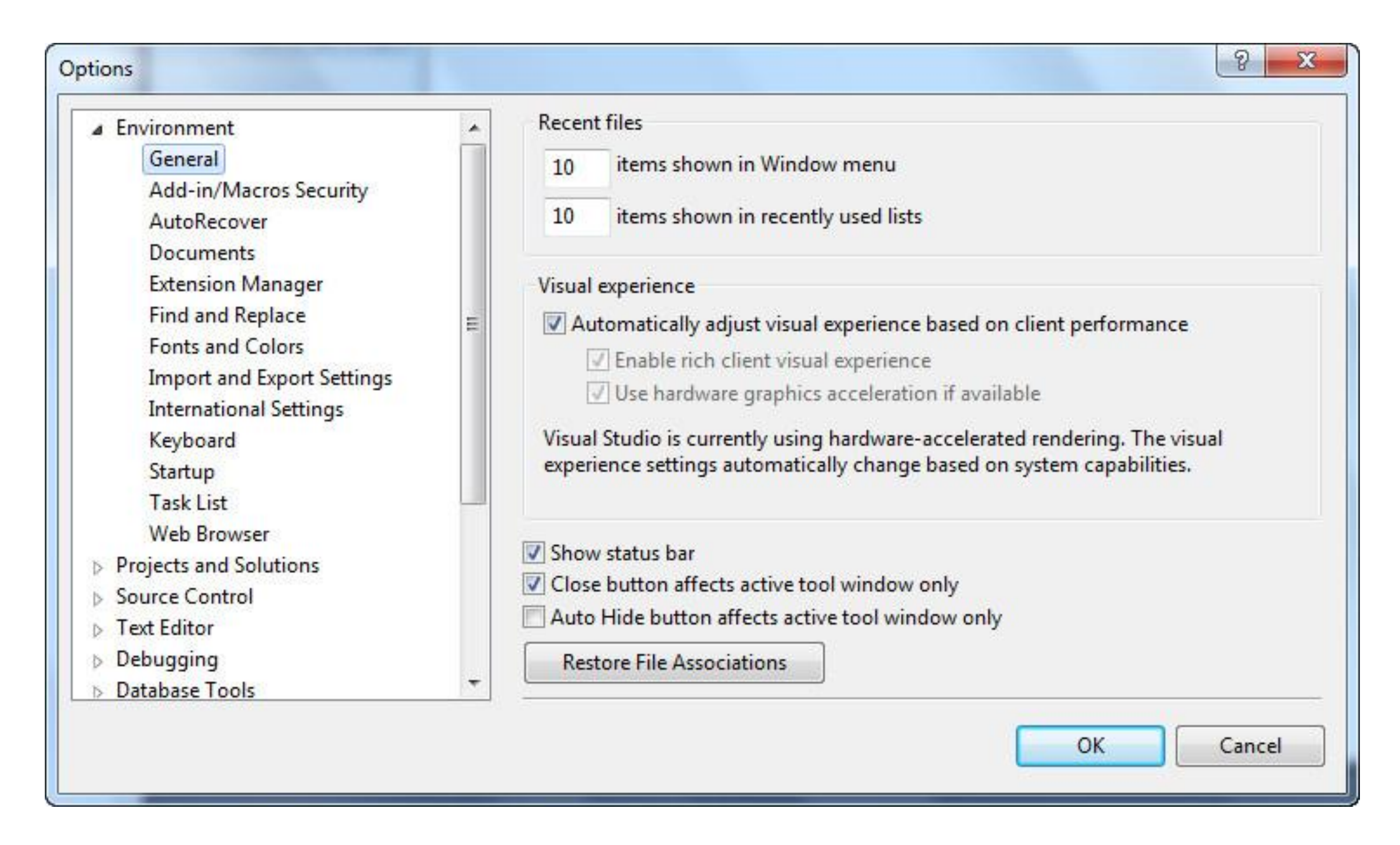
عند تشغيل بيئة التطوير **Visual Basic.Net** لاول مرة فانك ستلاحظ ظهور نافذة **Start Page** تستطيع من خلال هذه الشاشة استئناف العمل على اخر المشاريع المنجزة مؤخراً وكذلك بامكانك انشاء مشاريع جديدة كل هذا عن طريق هذه النافذة الظاهرة معنا من خلال (الشكل ٢-١)



شكل (٢-١) نافذة **Start Page** في بيئة التطوير **VB.Net2010**

نافذة الخيارات Options

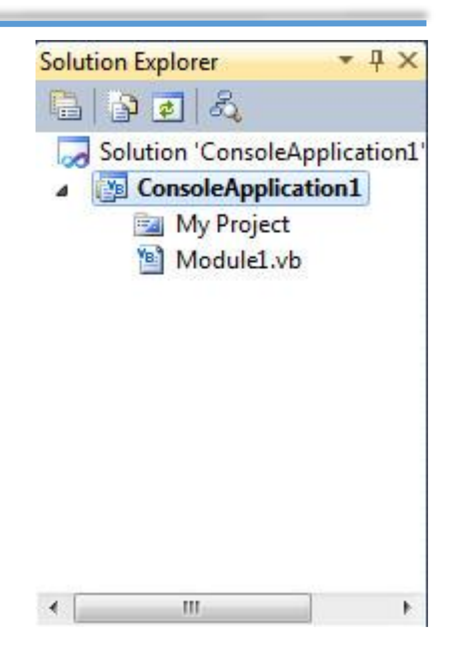
عند اختيارك للامر options من قائمة Tools يؤدي الى ظهور (الشكل ٣-١) فيمكنك عزيزي القارئ القيام بالعديد من اعداد عشرات الاوضاع والخيارات كاعداد بيئة التطوير ، محرر الشفرات ، المترجمات compilers وكذلك قواعد البيانات Database ...الخ من خيارات الاعداد .



شكل (٣-١) نافذة Options في بيئة التطوير VB.Net2010

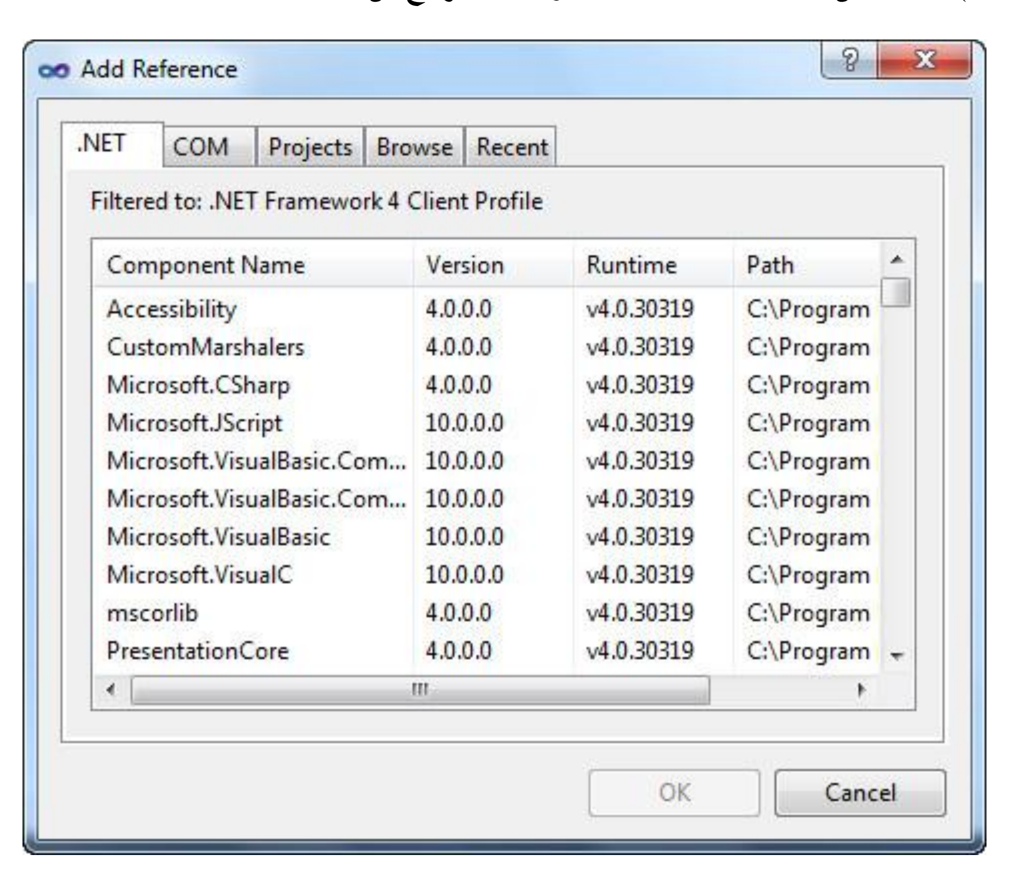
نافذة عرض الفئات Class View :

عزيزي القارئ تلاحظ من خلال هذه النافذه ظهورها على شكل شجرة مرتبة بنودها ، والغرض من هذه النافذه هو عرض فئات المشروع Project Classes ولقد اوضحنا لكم النافذة (بالشكل ٤-١) فكل الفئات التي تظهر في نافذة عرض الفئات هي كلها التي تضيفها انت عزيزي القارئ الى مشروعك الحالي فلن تظهر هذه الفئات في اي مشروع اخر تقوم بانشاؤه من تلقاء ذاتها .



شكل (٥–١) نافذة عرض الفئات **Class View**

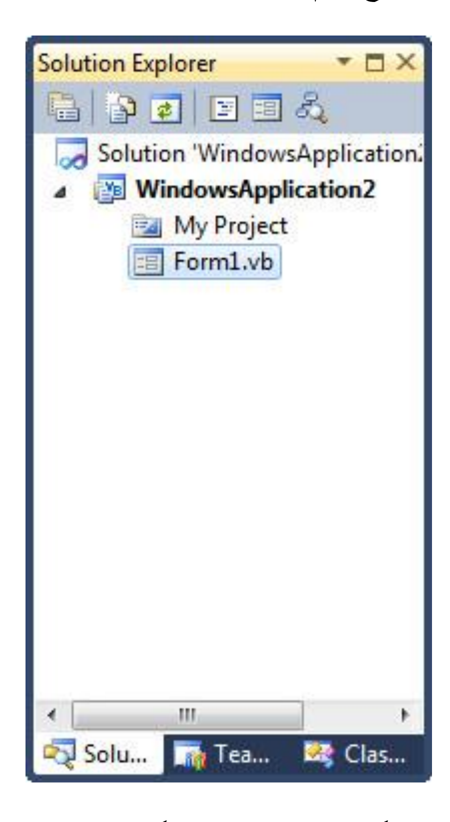
عزيزي القارئ يمكنك عرض نافذة اضافة المراجع Add Reference من خلال النقر بالزر الايمن للماوس على مشروعك المتواجد في نافذة **Solution Explorer** ومن ثم اختيار Add Reference ونوضح لكم شاشة اضافة المراجع من خلال (الشكل ٦–١)فيمكنك من خلال هذه الشاشة اضافة وحذف المراجع من هذه النافذة .



شكل (٦–١) نافذة اضافة المراجع Add Reference

نافذة مستكشف الحل Solution Explorer

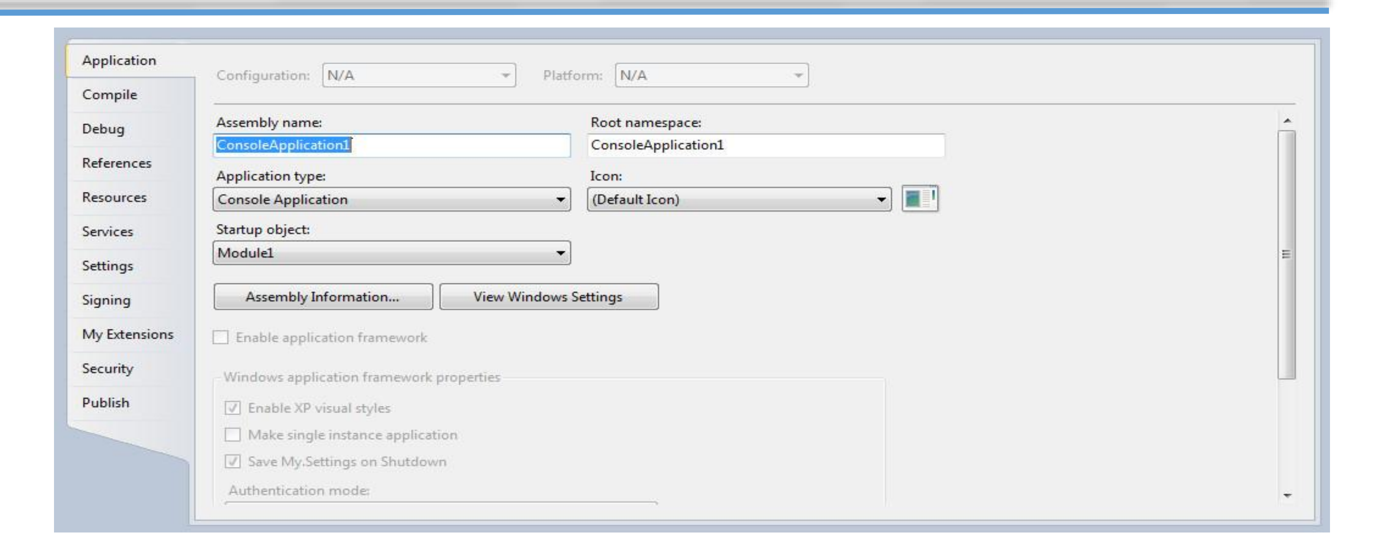
نافذة مستكشف الحل هي احدى نوافذ بيئة التطوير Net. والغرض من هذه النافذه هو عرض ملفات المشروع الحالي والمشاريع الاخرى ونوضح لكم قائمة مستكشف الحل من خلال (الشكل ٧-١)



(الشكل ٧-١) نافذة مستكشف الحل Solution Explorer

نافذة خصائص المشروع Project Property Pages

عزيزي القارئ من خلال هذه النافذة بامكانك اجراء عمليات الاعداد والتعديل النهائية على خصائص مشروعك ويمكنك الوصول الى هذه النافذة من النقر بالزر الايمن للماوس على مشروعك المتواجد بنافذة مستكشف الحل ومن ثم اختيار **Properties** وسنوضح لكم هذه الشاشة من خلال (الشكل ٨-١) .



(الشكل ٨-١) نافذة خصائص المشروع

نافذة الخصائص Property

عزيزي القارئ هذه النافذة تعتبر مهمة لك بالنسبة كا مبرمج Visual Basic.Net لانها تمكنك من التحكم بخصائص النماذج والادوات المضافة فيها في المشروع ك(الخطوط،ولون الخطوط،وخلفية النموذج ، واسم النموذج/الاداة ،موقع النموذج ، حجم النموذج/الاداة .....الخ) من الخصاص الاخرى كما انه يوجد تشابه في بعض هذه الخصائص من اداة لاخرى وسنوضح لك حالياً طرق تعيين الخصائص

توجد طريقتين لتعيين خصائص النماذج والادوات :

- الطريقة اليدوية

وهي الطريقة الاكثر سهولة في أن يقوم المبرمج بعملية تغيير خصائص النموذج أو الادوات المضافه اليه من خلال الذهاب الى نافذة الخصائص والقيام بتغيير تلك الخصائص من خطوط ، ولون خطوط ، وخلفية .....الخ

- التحكم بالخصائص من خلال الاكواد البرمجية

وهي الطريقة الاكثر اهمية والاكثر تعاملا معها في تصميم البرامج تحت بيئة Net. وحيث انها تكسب المبرمج مهارة عالية جداً فمن خلالها يقوم المبرمج بادخال اكواد برمجية في نافذة تحرير الاكواد تقوم بعملية التحكم بخصائص النموذج وتلك الادوات.

ملاحظة : عزيزي القارئ يوجد هناك فصل يتكلم عن خصائص الادوات والتعامل معها ، اضافة الى امثلة تمارين مكثفة بهذ الجانب.

القائمة الرئيسية

سنوضح لك عزيزي القارئ بشكل مختصر عن محتويات القائمة الرئيسية وهي تتكون من العديد من القوائم الرئيسية والفرعية .

القائمة File

ومايتواجد من خلال هذه القائمة من اوامر تجدها عزيزي القارئ متعلقة بمشروعك الحالي الذي انت تعمل عليه وتمكنك من فتح مشاريع اخرى واضافة ملفات اليها وحفظ مشروعك الحالي ....الخ من الاوامر .

القائمة Edit

من خلال هذه القائمة يمكنك تنفيذ العديد من الاوامر كـ النسخ واللصق والقص اضافة الى ادوات اخرى .

القائمة View

تتعلق هذه القائمة بالقيام بعملية عرض واخفاء العديد من النوافذ في بيئة **Net**.

القائمة Project

يوجد في هذه القائمة العديد من الاوامر الهامة المتعلقة بمشروعك الحالي والتي من خلالها بامكانك اضافة عناصر وملفات كما انه با مكانك ايضا التحكم باولوية المشاريع المتعددة في الحل الواحد من خلال الامر Set As Startup Project .

القائمة Build

من خلال هذه القائمة بامكانك القيام بعملية ترجمة Compiling اما بخصوص الامر Configuration Manager فهو يقوم بعملية تحديد الاعدادات المسبقة الحفظ للمترجم .

القائمة Debug

لحظة تصميم البرنامج تكون عدد عناصر هذه القائمة لا تتجاوز 9 أوامر، ولكن عند التنفيذ سيمتد هذا العدد ليصل إلى 13 أمر) بعضها غير مفعل(، تجد أوامر التنفيذ والإيقاف النهائي والمؤقت كما تحتوي هذه النافذة، على جميع وظائف التنقيح في البرنامج **Debugging** .

القائمة Tools

تحتوي على أوامر إضافية مختلفة الوظائف، كما يمكنك جعلها منصة لتشغيل برامج اخرى تستخدمها بشكل متكرر عن طريق اختيار الأمر **Externals Tools** وبالنسبة للإضافات **Add– Ins** فيمكن الوصول لها عن طريق اختيار الأمر **Add–In Manager**.

القائمة window

لاتعليق ..

القائمة help

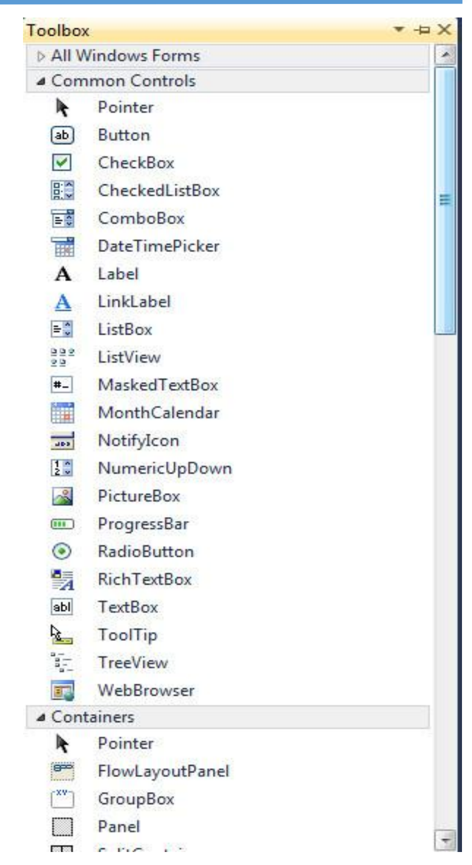
لاتعليق ..

اشرطة الادوات Toolbars

تعتبر اشرطة الادوات هي عباره عن اشرطة تحتوي على العديد من الاوامر ذاتها الموجودة في القائمة الرئيسية وبامكانك اضافتها أو حذفها من خلال النقر بزر الفارة الايمن على شريط الادوات ومن ثم اختيار الامر Customize.

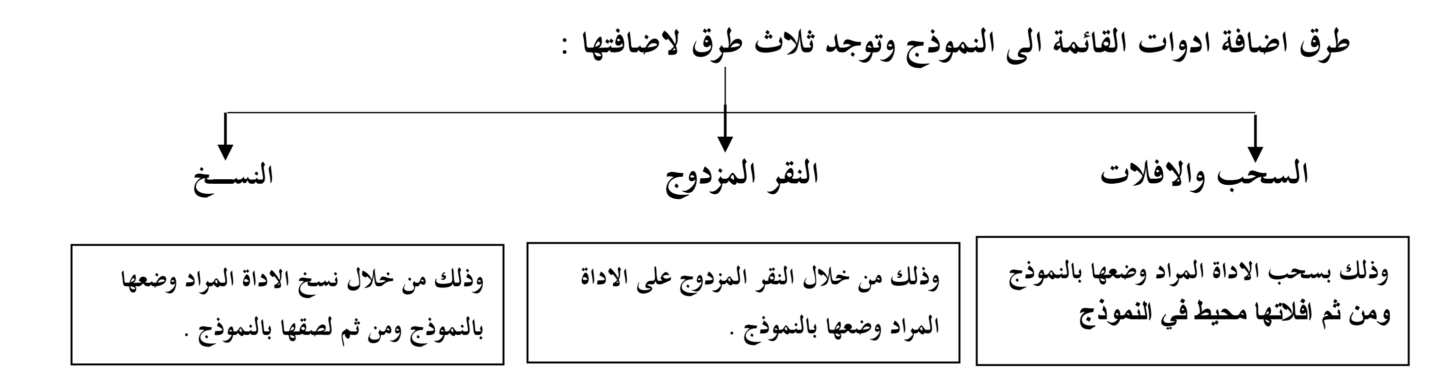
صندوق الادوات Tool Box

عزيزي القارئ سنوضح لك بشكل مفصل عن صندوق الادوات **Tool Box** كما تراه في بيئة **Net**. يوجد صندوق ادوات في الجهة اليسرى كما تلاحظون صندوق الادوات في (الشكل ٩-١) :



(الشكل٩-١) لصندوق الادوات Toolbox

طرق اضافة ادوات القائمة الى النموذج وتوجد ثلاث طرق لاضافتها :

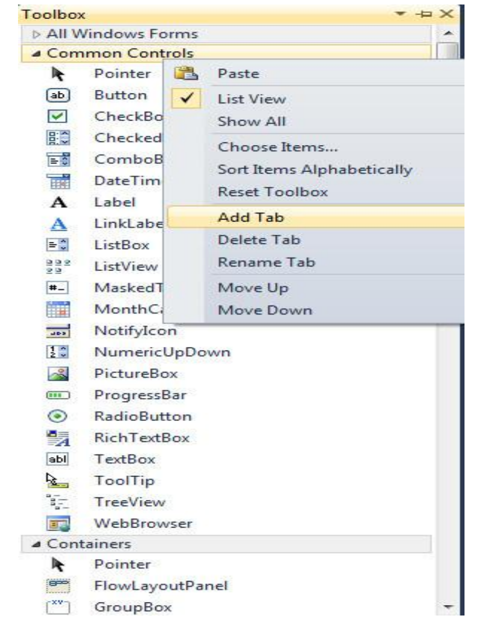


ملاحظة : لاضافة اداة النموذج لابد أن يكون النموذج أن يكون في وضعية التصميم وليس بوضع محرر الكود .

كما تتيح بيئة لك **Net**.القيام بعملية اضافة صندوق جديد الى صندوق الادوات أو حذف أو تعديل بحيث تقوم بعملية التنظيم بشكل رائع للادوات .

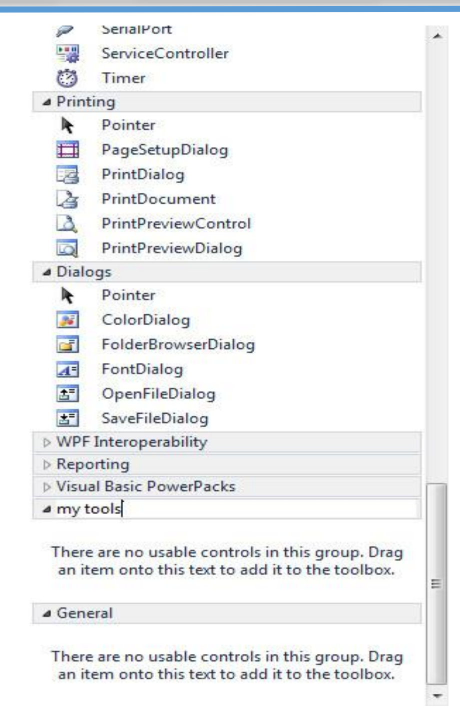
ولاضافة شريط **Tap** قم بالتالي :

- النقر بزر الفارة الايمن على صندوق الادوات لتظهر لك قائمة فرعية ومن ثم تختار منها **Add Tab** كما (بالشكل١٠-١) :



(الشكل١٠-١) يوضح طريقة اضافة شريط ادوات فرعي ضمن صندوق الادوات **Toolbox**

- ستلاحظ أن شريط الادوات يقوم بفتح مربع نص لك باسفل صندوق الادوات يتيح لك امكانية تسمية الشريط الجديد الذي ترغب باضافته لصندوق الادوات كما (بالشكل ١١-١) :



(الشكل١١-١) يوضح ويظهر مربع النص الذي من خلاله بتسمية شريط الادوات باسم "**my tools**"

كتابة برنامجك الاول

عزيزي القارئ يبدوا وكانك في شوق ولهفة لتبداء لمساتك الاولى بانشاء برنامجك الاول في بيئة Visual Basic.Net وانت تعلم أن الكتاب تم تاليفه وانشاء شروحاته في ظل بيئة Visual Basic.Net وهذا الاصدار يعتبر اصدار منقح ورائع جدا سترى وتلاحظ اولى خطواتك بانشاء برنامجك البدائي الان .

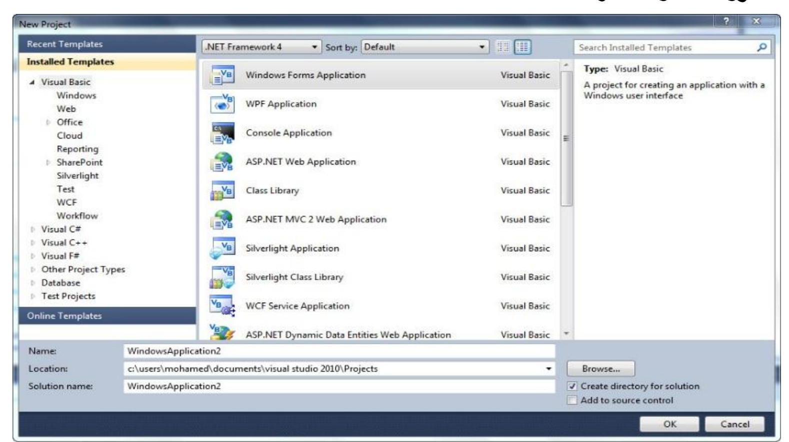
الحلول والمشاريع

قبل أن نشرع بالبدء باول خطواتنا بكتابة البرنامج الاول في بيئة Net. يلزم عليك عزيزي القارئ أن تفهم بعض المصطلحات الاساسية هنا

- الحل Solution : هو عبارة عن حاوي لعنصر أو مجموعة من المشاريع projects والمشروع هو البرنامج الذي تود انت انشائه ويحتوي هذا المشروع على ملفات المشروع project files .

عزيزي القارئ سنوضح لك الخطوات اللازمة للبدء بانشاء برنامجك الاول :

- الذهاب الى القائمة File -> ومن ثم اختيار الامر New -> ثم اختار منها الامر Project .
- ستفتح لك نافذة انشاء مشروعك الجديد وبها العديد من الخيارات المرتبطة بنوعية مشروعك الجديد الذي ترغب بانشائه فهناك مشاريع Net. بكافتها (Visual Basic.Net,vc#.Net,vf#.Net ....الخ وكل بيئة تطوير معينة تحتوي على انواع كثيرة من المشاريع فلنفترض ان برنامج الاول سيكون باستخدام بيئة التطوير vb.Net يجب عليك أن تختار Visual Basic.Net من نافذة المشروع الجديد ستلاحظ انه ظهرت لك مجموعة من المشاريع تحت بيئة التطوير vb.Net منها (windows forms application , wpf application , console application .....الخ) نحن سنفترض أن مشروعنا الاول سيكون من نوع windows forms application اختر هذا النوع من المشاريع وقبل أن تضغط على الزر ok تمهل عزيزي القارئ ولاحظ اسفل النافذة ستجد فيها موضحا اسم مشروعك الذي ستنشئه انت Name فليكن اسم المشروع الاول هو كما واضح معنا (بالشكل ١٢-١) وستجد اسفل مربع الاسم مربع نص اخر هو المسار location وهو يقصد به موقع المجلد الذي ستحفظ فيه مشروعك الجديد واسفل مربع المسار ستجد مربع نص اخر وهو اسم الحل الذي سيحتوي مشروعك او مشاريعك الجديدة solution name .



(الشكل ١٢-١) نافذة انشاء المشروع الجديد

انواع المشاريع

عزيزي القارئ تمكنك بيئة التطوير في لغة البرمجة Visual Basic.Net من دمج اكثر من مشروع في مشروع واحد فيمكنك اضافة مشروع جديد من نوع تطبيقات قياسية windows forms application ودمجه مع مشروع من نوع ادوات تحكم مثلا user controls وكذلك مع العديد من المشاريع المتواجدة في بيئة Visual Basic.Net والتي لاحظتها في نافذة New Project واليك عزيزي القارئ نبذة مختصرة عنها :

- **Windows Application**

هذا النوع من المشاريع يشبه كثيرا تطبيقات windows القياسية سنخصص عزيزي القارئ الفصل السادس من الجزء الثاني للكتاب لمناقشة وايضاح تطوير تطبيقات windows وبشكل مكثف بهذا الجانب.

- **Class Library**

طبعاً هذا النوع من المشاريع يحتوي على مكتبة تحتوي على العديد من الفئات التي يمكنك الاستفادة منها في برامج اخرى كما يمكنك ترجمتها وتحويلها الى ملفات من نوع dll. والاستفادة منها في مشاريع اخرى وسنخصص لكم مجالا لشرح هذا النوع من المشاريع في الجزء الثالث من كتابنا.

- **Windows Control Library**

يمكنك هذا النوع من المشاريع من انشاء ادوات تحكم user controls يمكنك الاستفادة منها في مشاريعك من النوع الطبيقات القياسية Windows Application وسنخصص لكم مجالا لشرح هذا النوع من المشاريع في الجزء الثالث من كتابنا.

- **ASP .NET Web Application**

يمكنك من خلالها من انشاء مشاريع ASP.NET بحيث تعمل في جهة الخادم server ويتم عرض صفحاتها من خلال العملاء clients باحد المتصفحات Browsers وسنخصص لكم مجالا لشرح هذا النوع من المشاريع في الجزء الثالث من كتابنا.

- **ASP .NET Web Service**

هذا النوع من المشاريع يسهل عليك عملية تبادل البيانات عبر الانترنت عن طريق استخدام بروتوكولات XML و HTTP القياسية دون الحاجة إلى تطوير المكونات الموزعة DCOM وسنخصص لكم مجالا لشرح هذا النوع من المشاريع في الجزء الثالث من كتابنا.

- **Web Control Library**

هذا النوع من المشاريع يمكنك من انشاء ادوات التحكم ولكنها هنا خاصة للعرض على صفحات html وهي تشبه الى حد كبير مشاريع user controls ولكنه ستعرض من خلال متصفحات الويب وسنخصص لكم مجالا لشرح هذا النوع من المشاريع في الجزء الثالث من كتابنا.

- **Console Application**

يعمل هذا النوع من المشاريع تحت بيئة Dos ويعتبر هذا النوع من التطبيقات مناسبا الى حد ما للمبتدئيين وكذلك لمن يرغبون بتطوير مشاريعهم من هذا النوع تحت بيئةNet.

- **Windows Services**

يعد هذا النوع من المشاريع القياسية windows ومن مزاياه انه يعمل في الخلفية Background منذ بداية تحميل نظام التشغيل حتى اغلاق جهاز الكمبيوتر وسنخصص لكم مجالا لشرح هذا النوع من المشاريع في الجزء الثالث من كتابنا.

- **Empty Project**

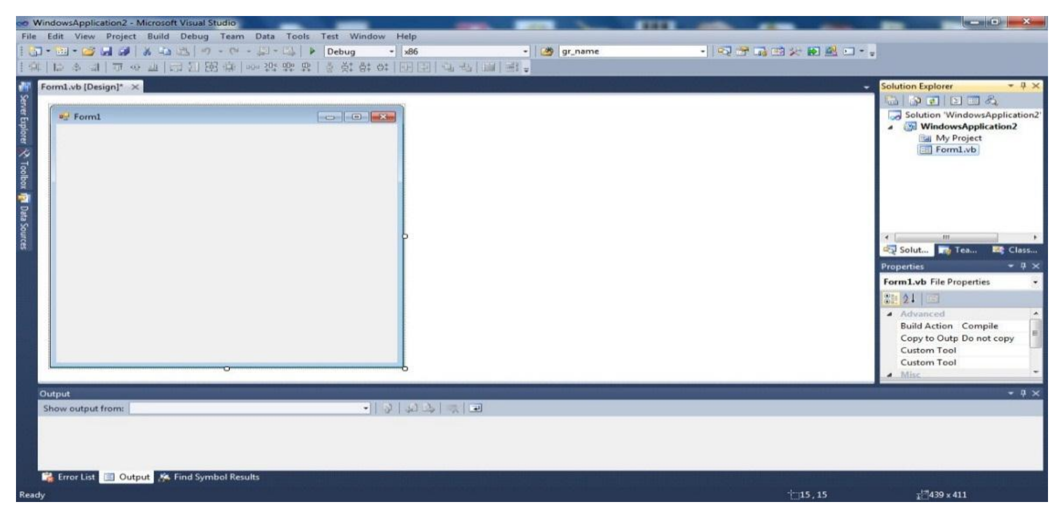
لاتعليق!!

- **Empty Web Project**

لاتعليق!!

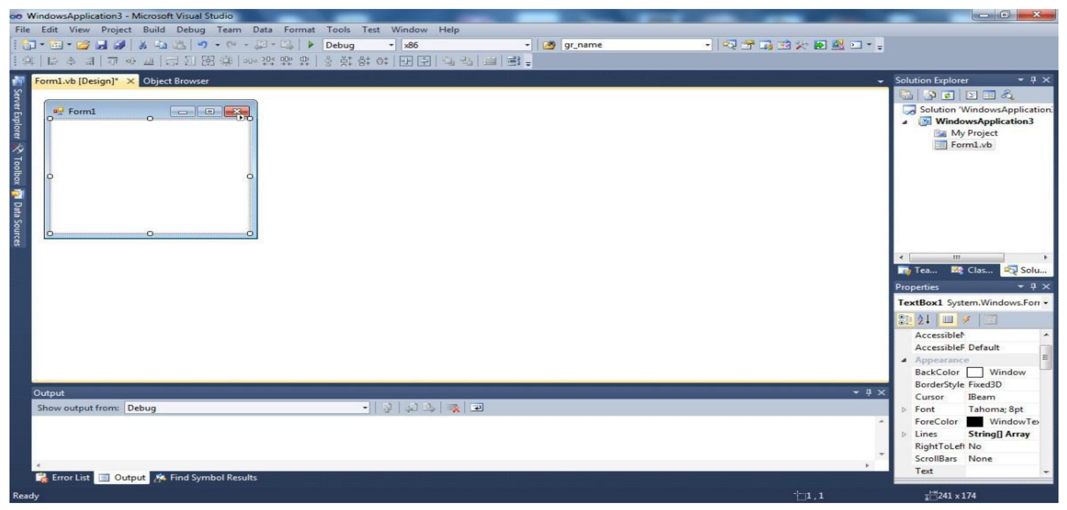
بناء برنامجك الاول

بامكانك عزيزي القارئ ان تبداء ببناء برنامجك الاول بمختلف الطرق وبامكانك اختيار نوع المشروع الذي سيكون بمثابة برنامجك الاول في
بيئة تطوير Visual Basic.Net عزيزي القارئ سنوضح لك في هذا الفصل الخطوات الاولية لكيفية بناء برنامجك الاول وتنفيذه فقط ولن نتوغل اكثر من ذلك في قواعد مفردات اللغة ، فسيكون مثالنا هنا عن بناء برنامجك الاول من نوع windows application من نوع التطبيقات القياسية فبعد أن تعرفنا سابقاً عن نافذة انشاء مشروع جديد لنعود اليها حالياً ونقوم بالذهاب الى نافذة انشاء مشروع جديد من نوع تطبيقات قياسية windows application بعد الخطوة السابقة باختيار نوع مشروعنا الجديد وبعد الضغط على الزر Ok تظهر لنا نافذة العمل على مشروعنا الحالي الذي اخترناه كماهو موضح (بالشكل ١٣-١)



(الشكل ١٣-١) نافذة بيئة التطوير بعد اختيار المشروع الجديد

لنجعل برنامجك الاول عبارة عن برنامج يحتوي على مربع نص واحد فقط يقوم بعرض رسالة ترحيبية في برنامجك الاول في بيئة Net. ولذلك نقوم باضافة اداة مربع نص Textbox من شريط الادوات الجانبي Tools Box ليصبح شكل النموذج كما هو في (الشكل١٤-١) ونقوم مثلا بعملية اعادة تسمية لعنوان النموذج من خلال الخاصية – الشكل Text 11 المتواجدة في نافذة الخصائص properties وتغييرها الى "برنامجك الاول"



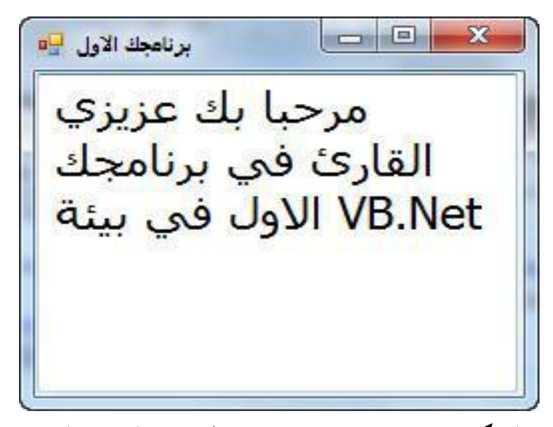
(الشكل ١٤-١) نافذة اضافة مربع النص Textbox الى نافذ النموذج Form1

يمكنك الوصول الى نافذة شفرة البرنامج وكتابة الشفرة الترحيبية التي ستظهر في مربع النص Text Box باكثر من طريقة فيمكنك على سبيل المثال الذهاب الى نافذة الحل solution explorer ثم النقر على النموذج المتواجد في النافذة form1 بالزر الايمن للفارة ومن ثم اختيار view code عرض الشفرة ومن ثم كتابة الشفرة الاتية:

Visual Basic.Net	code

```
Public class form1 Private sub form1_load (by Val sender as system. Object, by Val
e as system. Event rags) handles my base. load
Textbox1.text = " مرحبا بك عزيزي القارئ في برنامجك الاول في بيئة vb.Net"
End sub
End class
```

بعد ان انتهينا من كتابة شفرة برنامجك الاول بعدها ننتقل الى عملية التنفيذ من خلال الضغط على المفتاح F5 ستلاحظ بعدها ظهور برنامجك كما) بالشكل ١٥-١ (



(الشكل ١٥-١) برنامجك الاول بعد القيام بالتنفيذ

الترجمة والتوزيع

تعتبر عملية الترجمة **Compiling** هي الاخطوة الاهم لترجمة مشروعك وتوزيعه الى ملف تنفيذي **exe**. أو **dll**. ويمكنك القيام بها من خلال عملية الاعداد المسبق لها **Configurations** ومنها يمكنك اختيار الاعداد **Debug** كـ اعداد مبدائي وهو كما هو متواجد في شريط العنوان في بيئة **Visual Basic.Net** وسنوضح ذلك من خلال (الشكل ١٦-١):

Debug
Release
Configuration Manager...

(الشكل ١٦-١) لاختيار **Debug** الامر

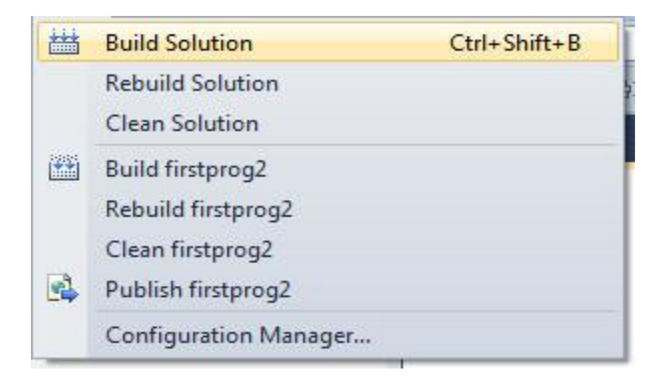
اما بشان الاعداد **Release** فهو يتم اختياره بحال كان مشروعك بمرحلته الاخيره يتم اختيار هذا الاعداد قبل عملية التوزيع للبرنامج ويمكنك ايضا تحرير هذه الاعدادات من خلال **Configuration Properties** ، اما عن المرحلة الاخيرة عملية التوزيع هذه المرحلة تقوم بها بحال رغبت بتوزيع برنامج الى اجهزة اخرى يجب أن يتم ارفاق نسخة من مكتبة اطار عمل **Net Framework.** مع البرنامج حتى تعمل .

عزيزي القارئ سيتضح لك الكثير عن اطار العمل **Net Framework.** في الفصول القادمة أن شاء الله وسيتم ايضاحها بشكل اكثر .

كيف تبني مشروعك الاول ؟

عزيزي القارئ يمكنك القيام ببناء مشروعك الحالي بعد أن قمت بتجربته من خلال التشغيل **F5** يمكنك بعد ذلك بناء مشروعك الاول مثلا بامتداد **EXE**. وذلك من خلال القائمة **Build** قم باختيار **Build Solution** أو من خلال الضغط على **Ctrl+shift+B**

انظر (الشكل١٧-١) ولاحظ طريقة بناء برنامجك الاول :



(الشكل ١٧-١) طريقة القيام بعملية بناء مشروعك الاول

ستلاحظ أن البرنامج يقوم بتجميع وترجمة ذلك وانتاج ملف تنفيذي لمشروعك بصيغة **EXE**. ولكي تعرف مكان تواجد ملفات المشروع التي بنائها ماعليك الا أن تذهب الى مسار انشاء المشروع ثم المجلد **bin** ومنه تذهب الى المجلد **Debug** وستجد فيه ملف مشروعك بصيغة **EXE**.

الفصل الثاني

لغة البرمجة

سنتطرق في هذا الفصل على العديد من الامور الهامة التي يجب على المبتدئيين في لغة البرمجة Visual Basic.Net اتقانها وفهمها لكي تبنى عليها ارضية صلبة من المعلومات المتسلسلة والتي تمكن المبرمج من التعامل مع لغة البرمجة بشكل احترافي وذلك من خلال التعرف على :

- **مقدمة عن الوحدات البرمجية Modules**
- **تطبيقات console application**
- **المتغيـــرات والثوابـــــت**
- **المصفوفات**
- **التفــــــرع والتكـــــرار**
- **مقدمة عن تصحيح الاخطاء**

لغة البرمجة

عزيزي القارئ لن نقوم بتكثيف العمل على تطوير تطبيقات Windows منذ الوهلة الاولى في هذا الفصل فيعتبر الامر مبكرا للبدء بذلك ، عزيزي القارئ اذا .اردت أن تفهم لغة البرمجة Visual Basic.Net ماعليك الا أن تفهم الاساسيات لكي تجيدها جيدا وأن معظم الاوامر والشفرات البرمجية مرتبطة وهي مهمة جدا البدء بها من خلال مايسمى بالوحدات البرمجية Modules التي تحمل ملف مستقل يحمل الامتداد vb.سنخصص في هذا الفصل لكل ماهو يهم المبتدئيين في التعامل مع كافة العمليات الاخرى من اجراءات ومتغيرات وثوابت ومصفوفات ودوال ....الخ في ظل منصة تطبيقات قياسية وبشكل بدائي وسنقوم بشرح مجمل عليه بشكل مكثف جدا .

الوحدات البرمجية Modules

عزيزي القارئ ماهي الوحدة البرمجية ؟ وماهي مكونات شفرتها ؟ واين يتم كتابة الشفرات فيها ؟

- الوحدة البرمجية Module : هي عبارة عن وحدة تحتوي على جميع الاجراءات التي نريد استخدامها في المشروع ويتم استدعاء هذه الاجراءات من داخل أي نموذج بالمشروع .
- تتكون شفرتها الاولية من الاتي :

Visual Basic.Net	code

```
Module Module1

Sub Main()

End Sub
End Module
```

عزيزي القارئ كل نموذج يحتوي بداخله على عدة اجراءات معرفة بداخله فسوف تجد صعوبة في تذكر النموذج الموجود به الاجراء الذي نريد استخدامه ولهذا فان لغة البرمجة Visual Basic.Net تسمح لنا بانشاء مساحات خاصة من الكود تسمى بالوحدات البرمجية Modules ونستخدم هذه المساحة لاحتواء وتنظيم كل الاجراءات التي نريد استخدامها فبدلا من انشاء الاجراءات داخل النماذج بصورة متفرقة فيمكنك انشاء وحدة برمجية Module تحتوي على جميع الاجراءات التي نريد استخدامها بالمشروع ويتم استدعاء هذه الاجراءات من خلال أي نموذج بالمشروع والوحدة Module عبارة عن ملف منفصل ياخذ الامتداد vb. وهناك العديد من الخواص التي تميز الوحدات Modules فهي :

- تشبه نافذة محرر الكود ولاتحتوي على مكونات رسومية .
- كما انها غير مرتبطة باي نموذج وايضا لايمكن اضافة أي من عناصر التحكم داخل هذه الوحدات .

- كما ان الكود المسموح اضافته داخل هذه الوحدات هو تعريف المتغيرات **variable** والاجراءات **procedures** والوظائف **Functions** فقط .
- وايضاً تحتوي على جزء التعريف العام فقط **General Deceleration** .

فلايمكنك كتابة الشفرة خارج نطاق الوحدة البرمجية بمعنى لايمكنك كتابة الشفرة فوق السطر **Module Module1** ولاتحت السطر **End Module** بمعنى أن الكود البرمجي يجب أن يكون مابين الاجراء **Sub Main** واقفاا الاجراء **End Sub** وبامكان مشروعك أن يحتوي اكثر من وحدة برمجية بشرط لاتتعارض الوحدات البرمجية **Modules** في مسمياتها كما انه يجب الاتزيد عدد حروف مسمى الوحدة البرمجية **Module** أن لايزيد عدد حروف المعرف فيها عن ١٦٣٨٣ حرف وانا اعتقد عزيزي القارئ انك لست بحاجة لتكتب هذا الكم من العدد ! ، وكذلك لايمكنك من استخدام الكلمات المحجوزة **Keyword** لتعريف معرف جديد لنبداء معكم خطوة بخطوة في انشاء مشروع جديد من نوع **windows application**

تطبيقات Console Application

بعد أن تحدثنا عن الوحدات البرمجية في الموضوع السابق سنقوم بتخصيص تطبيق محدد بشكل مبدائي للتعامل معها وليكن هو تطبيق **console application** وهذا النوع هو مايتلائم بالتعامل مع **Module** بشكل يتناسب مع المبتدئيين وسنوضح لكم لاحقاً بفصول قادمة التعامل بشكل اكبر ومفصل مع الوحدات البرمجية **Modules**هذ النوع من التطبيقات ليس له واجهة تطبيق ولكنه عبارة عن **Module** ، والكلمة **Main** اسم البرنامج الفرعي ،يجب التعرف على الاساليب المستخدمة لقراءة مدخلات التطبيق وهي :

Console .Read line	يستخدم لقراءة سطر الكتابة من وحدة الادخال القياسية
Console .Read	يستخدم لقراءة الحرف التالي من وحدة الادخال القياسية

اما الاساليب المستخدمة لعرض مخرجات التطبيق :

Console .Write line	عند استخدامه فان جملة الادخال أوالاخراج يتم عرضها في السطر التالي لسطر المخرجات بهذه الجملة .
Console. Write	عند استخدامه فان جملة الادخال أو الاخراج يتم عرضها في السطر التالي لسطر هذه الجملة .

فعند كتابة اسطر الكود فان برنامج هذا الكود يتم عرضه بسطر واحد:

Console. Write ("Pc lab")

Console. Write Line ("Graphics")

Pc lab Graphics

وذلك لان الاسلوب **Console . Write** يخبر البرنامج فان يقوم بعرض ناتج جملة الاخراج التالية في نفس تسطر ، وعند كتابة اسطر الكود بهذه الطريقة:

Console. Write Line ("pc lab")

Console. Write Line ("Graphics")

أو بهذه الطريقة :

Console. Write Line ("pc lab")

Console. Write ("Graphics")

يكون الناتج كالتالي :

Pc lab

Graphics

وهذا لان الاسلوب Console. Write Lineيخبر البرنامج بعرض ناتج جملة الاخراج التالية في السطر التالي لهذه الجملة وسوف نلمس هذا الفرق عملياً بوضوح في الجزء التالي.

تمرين :

تعرفت في المثال السابق على تطبيقات Console application وسوف نقوم بهذا الجزء بانشاء تمرين للتعرف على كيفية التعامل مع مثل هذه التطبيقات فاذا اردنا على سبيل المثال انشاء برنامج لاجراء عملية قسمة رقمين وعرض ناتج القسمة فسوف نحتاج في بداية الامر لقراءة هذان الرقمان اولاً ثم اجراء عملية القسمة واظهار الناتج وكما ذكرنا من قبل انه لقراءة جملة من وحدات الادخال مثل لوحة المفاتيح نقوم باستخدام الاسلوب :

Console. Read Line

ولذلك لقراءة الرقم الذي نريد قسمته نكتب الجملة :

First = Console. Read Line

ثم الضغط على الزر Enter من لوحة المفاتيح .

تعمل هذه الجملة على قراءة الرقم الذي نريد قسمته من خلال لوحة المفاتيح وتخزين هذا الرقم في المتغير First وكما ذكرنا من قبل فان في الوضع الافتراضي للبرنامج يجب أن يقوم بتعريف جميع المتغيرات التي سوف نستخدمها داخل البرنامج ولذلك فانه سوف يظهر خط ازرق اللون متعرج تحت المتغير First فنقوم بتعريف هذا المتغير بنوع البيانات Double

Dim First as Double

وبنفس الطريقة نقوم بتعريف متغيرات اخراج لتخزين الرقم المقسوم عليه وناتج القسمة

Dim Second, Result as Double

ولقراءة الرقم المقسوم عليه ندخل الجملة

Second = Console. Read Line

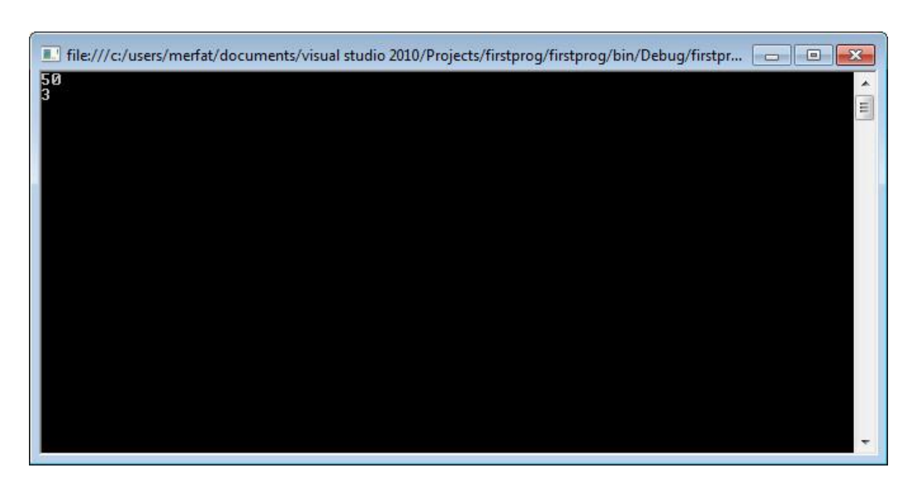
وبهذا فان الرقم الذي نريد قسمته يتم تخزينه في المتغير **First** والرقم المقسوم عليه يتم تخزينه في المتغير **Second** ولاجراء عملية القسم ندخل الجملة :

Result = First/Second

وهذه الجملة تقوم بقسمة الرقم الموجود في المتغير **First** على الرقم الموجود في المتغير **Second** ووضع ناتج القسمة في المتغير **Result** ولاظهار ناتج القسمة نقوم باستخدام الاسلوب فنكتب الجملة :

Console. Write line (Result)

وتعني هذه الجملة عرض قيمة المتغير **Result** والذي يحتوي على ناتج العملية ولرؤية الكود الذي قمنا بادخاله نقوم بالضغط على الزر **F5** كما في شاشة المحث (بالشكل ١-٢) :



(الشكل ١-٢) يوضح ظهور نافذة المحث كذلك القيام بعملية ادخال الرقم المقسوم ٥٠ والرقم المقسوم عليه ٣

فيتم فتح نافذة المحث التي تستخدم لقراءة المدخلات وعرض المخرجات فتدخل الرقم الذي نريد قسمته مثلا ٥٠ ثم نضغط على المفتاح **Enter** ثم ندخل الرقم المقسوم عليه مثلا ٣ ثم نضغط **Enter** نلاحظ أن الشاشة تختفي بشكل سريع بدون أن نتمكن من مشاهدة ناتج القسمة ولكي نتمكن من مشاهدة ناتج القسمة نقوم باضافة سطرين كود وهما :

Console. Write line ("press any key to exit?")

Console. Read line ()

ويجب ملاحظة أن كل تطبيق **console** يجب أن يحتوي هاتان الجملتان في نهاية الكود لكي يتمكن المستخدم من مشاهدة ناتج البرنامج. والان نضغط على المفتاح **f5** لتشغيل البرنامج فيظهر لنا كما (بالشكل ٢-٢) :

ندخل الرقم الذي نريد قسمته ٥٠ ونضغط **enter**

ثم ندخل الرقم المقسوم عليه ٣ ونضغط **enter**

فيظهر ناتج عملية القسمة ونلاحظ انه تم اغلاق النافذة بعد ظهور الرسالة التي تطالب بالاغلاق .

```
file:///c:/users/merfat/documents/visual studio 2010/Projects/firstprog/firstprog/bin/Debug/firstpr...
50
3
16.6666666666667
press any key to exit?
```

(الشكل ٢-٢) يوضح اظهار ناتج عملية القسمة ومن ثم تظهر رسالة تطلب منك ضغط أي مفتاح للخروج من البرنامج

تمرين :

قم بانشاء برنامجك الاول باستخدام التطبيق **console application** يتضمن عملية قسمة رقمين وعرض نتيجة القسمة في متغير الناتج بحيث يقوم البرنامج باظهار رسالة اولية تطالب المستخدم بمحث الادخال بادخال الرقم الذي ترغب بقسمته وكذلك اظهار رسالة تطالب المستخدم بادخال الرقم المقسوم عليه ومن ثم عرض رسالة تتضمن جملة = **Result** ثم قم بعرض وتنفيذ هذا البرنامج ؟

الاجابة :

من نافذة تحرير الشفرة الخاصة بالبرنامج نقوم بالاتي :

- لاظهار الرسالة الاولية للمطالبة بادخال الرقم الذي تريد قسمته نقوم بادخال هذه الجملة قبل الجملة الخاصة بالرقم نريد قسمته:

Console. Write ("Enter First number?")

- لاظهار الرسالة التي تطالب بادخال الرقم المقسوم عليه نكتب الجملة الاتية قبل الجملة التي تطالب بادخال الرقم المقسوم عليه

Console. Write ("Enter Second number?")

- لاظهار الرسالة التي تطالب بعرض = **Result** نقوم بادخال هذه الجملة قبل الجملة التي تعرض ناتج القسمة

Console. Write ("Result=")

يصبح الشكل النهائي لكود البرنامج :

Visual Basic.Net	code

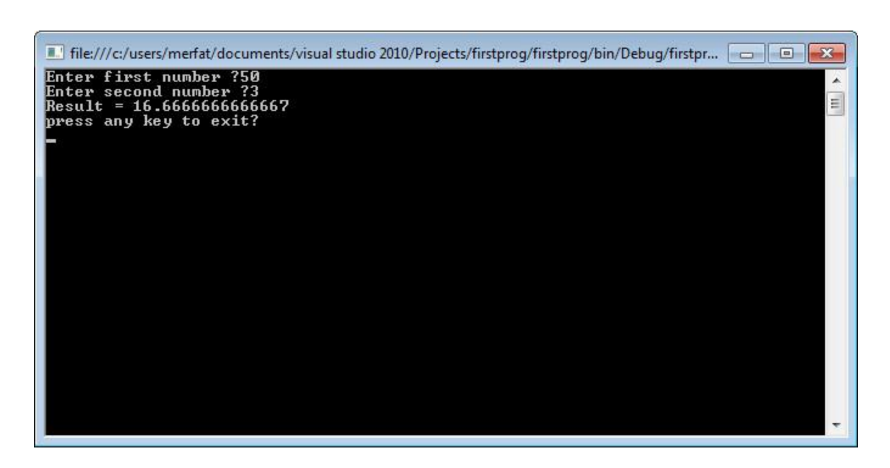
```
Module Module1

Sub Main()
Dim First as Double
Dim Second, Result as Double
Result = First/Second
Console. Write ("Enter First number?")
First = Console. Read Line

Console. Write ("Enter Second number?")

Second = Console. Read Line
Console. Write ("Result=")
Console. Write line (Result)
Console. Write line ("press any key to exit?")
Console. Read line ()
End Sub
End Module
```

كما تلاحظ تحقيق الكود في شاشة المحث من خلال (الشكل ٣-٢)



(الشكل ٣-٢) يوضح ظهور رسائل تطالب المستخدم بادخال البيانات

المتغيرات Variables

المتغيرات : هي اماكن لتخزين البيانات ، فيمكننا اجراء عملية متعددة داخل هذا لبرنامج ومن ثم يمكن لنا استخدام هذا البرنامج بصورة مستمرة بدلا من اعادة حساب هذه العملية الحسابية بشكل متكرر في المواضيع المختلفة للبرنامج ، تستخدم المتغيرات **variables** لتخزين الاتي :

Numbers	الارقام
Text	النصوص
Date	التاريخ
Times	الاوقات
....	بالاضافة الى بيانات اخرى

وتحتاج لغة البرمجة **Visual Basic.Net** الى معرفة نوع البيانات المستخدمة مع المتغير ويتم تحديد نوع البيانات للمتغير عن طريق نوع البيانات **Data Type** فنوع البيانات عن تطبيق للحجم وطبيعة البيانات التي يتم تخزينها بالمتغير وقبل اساخدام أي متغير داخل الكود فلابد من تعريفه عمل **deceleration** لهذا المتغير ومعنى التعريف أو عمل **deceleration** للمتغير هو حجز مكان بالذاكرة المؤقته **Ram** للمتغير لحفظ هذه البيانات وذلك بدلاً من استخدام عنوان المتغير في الذاكرة **Ram** للتعامل مع البيانات ويجب اتباع بعض القواعد عند تسمية المتغيرات ، مثل أن يبداء اسم المتغير بحرف

A, B, C................Z

كما يجب الا يتضمن اسم المتغير على أي مسافات خالية أو رموز خاصة **special characters** :

~ @ # $ % & { } : ; " ' , . ? ^ / * – +

ويجب الايزيد عدد الحروف للمتغير عن ٢٥٥ حرف ويجب الا يتكرر داخل البرنامج الفرعي الواحد .

الصيغة العامة لتعريف المتغيرات :

Dim Variable Name AS Data Type

حيث يبداء تعريف المتغير بالكلمة **Dim** متبوعة باسم المتغير **Variable Name** ثم الكلمة **AS** متبوعة بنوع البيانات التي تريد تعيينه لنوع المتغير ويجب ملاحظة أن انه يجب اختيار نوع البيانات التي نريد استخدامها المتغير بعناية لتفادي استخدام مساحات كبيرة

من الذاكرة بدون أي داعي وسوف نتعرف على انواع البيانات تفصيليا بالمواضيع القادمة ، فلتعريف متغير الاسم **X** بنوع البيانات **integer** فتكتب الجملة :

Dim X AS integer

ولتعريف متغير له الاسم **Y** بنوع بيانات **String** فتكتب الجملة :

Dim Y AS String

وتسمح لنا لغة **VB.NET** بتعريف اكثر من متغير بنفس الجملة فاذا اردنا على سبيل المثال تعريف المتغيرات **x , y** بنوع بيانات **integer** فتكتب الجملة :

Dim x, y as integer

واذا اردنا تعريف المتغير **x** بنوع **integer** والمتغير **y** بنوع بيانات **double** فتكتب الجملة :

Dim x as integer, y as double

ويمكننا تحديد قيمة ابتدائية ٥ لاحدى المتغيرات بطريقتين :

- الطريقة الاولى :

Dim x as integer

X=5

- الطريقة الثانية :

Dim x as integer = 5

ولتعيين اكثر من قيمة لاكثر من متغير بنفس الجملة :

Dim x as integer= 5, y as double =4.99

ويمكن تعريف المتغير بدون تعيين البيانات له وذلك بكتابة الجملة بهذه الصيغة :

Dim variable name

وبهذه الصيغة فان البرنامج يقوم بتحديد **object** تلقائياً لهذا المتغير وهذا النوع من البيانات اكثر انواع البيانات مرونة حيث يلائم جميع انواع البيانات المتاحة حيث يتم تعيين البيانات للمتغير حسب القيم التي يتم تصنيفها له ففي الجملة التي نراها :

Dim x

X = 5

يقوم البرنامج بتعيين نوع البيانات بشكل تلقائي **integer**

وعند كتابة هذه الجملة

```
Dim x
X = "Mohammed Hazim"
```

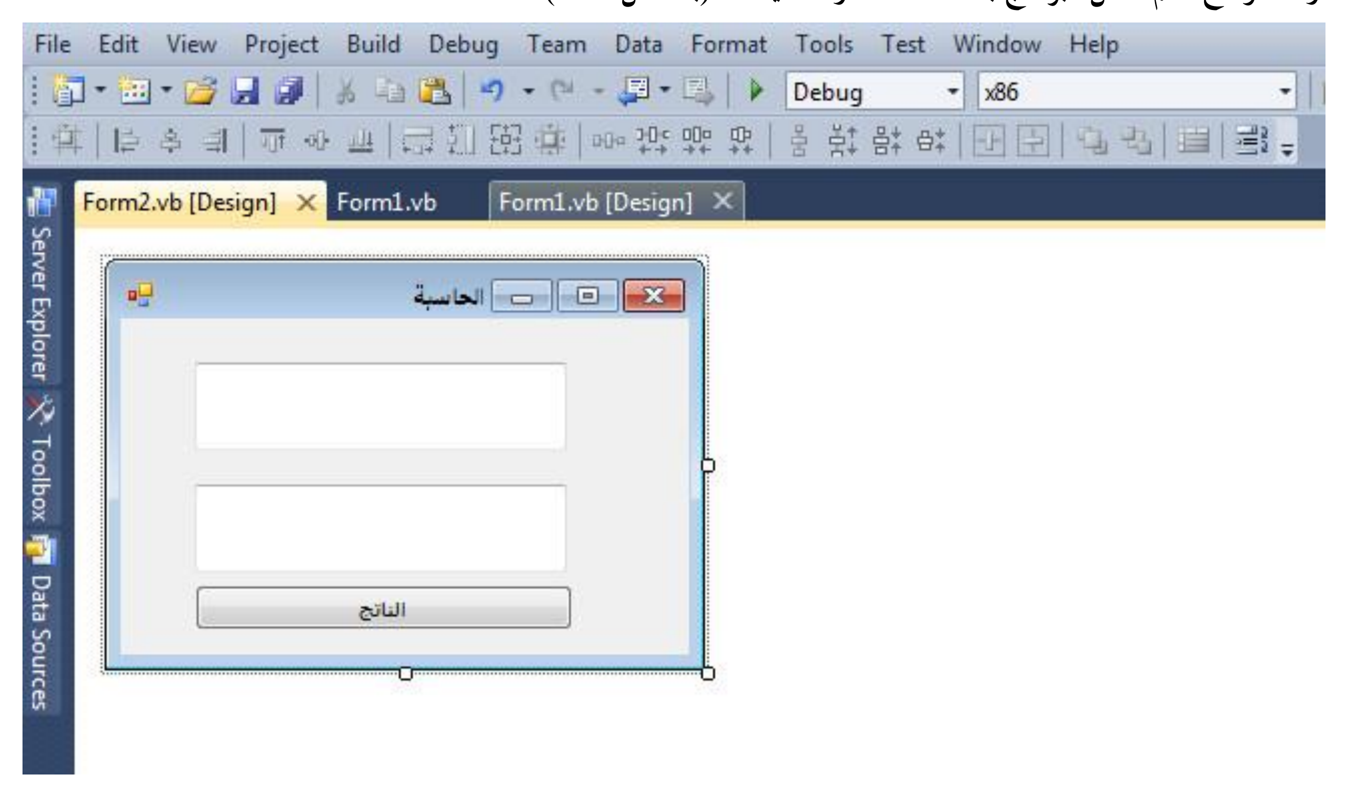
يقوم البرنامج هنا تلقائيا بتحديد نوع البيانات **string**

ولكن هذا النوع من البيانات ياخذ مساحة كبيرة من الذاكرة المؤقتة **ram** ولذلك لايفضل استخدامه لتعريف المتغيرات الا عند الظرورة .

تمرين ١

قم بانشاء برنامج يقوم باجراء عمليات حسابية بحيث يقوم البرنامج بكتابة العمليات الحسابية قم باضافة مربع نص **textbox** العدد ٢ وقم باضافة **button** العدد ١ ومن ثم اكتب الكود المتعلق بذلك داخل **Button1**

- اولا سنوضح لكم شكل البرنامج بعد اضافة الادوات اليه كما (بالشكل 4-٢)



(الشكل٤-٢) نافذة اضافة الادوات الى برنامج الحاسبة

- نوضح لكم الان الكود المتعلق باجراء العمليات الحسابية واظهارها في رسالة نصية كماتلاحظون (بالشكل ٥–٢)

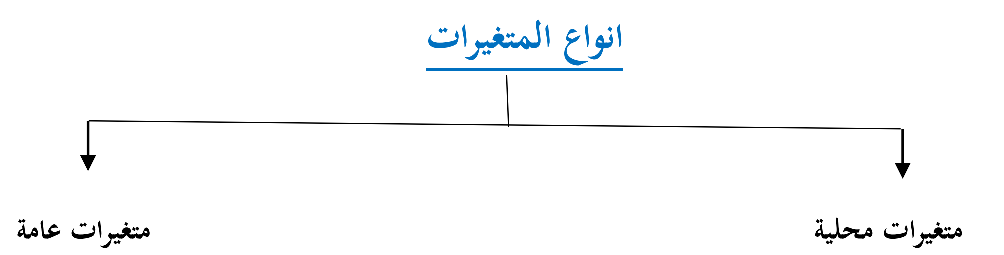
```
Public Class Form2

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Dim firsttext As Integer = TextBox1.Text
        Dim secondtext As Integer = TextBox2.Text
        Dim result As Integer = firsttext + secondtext
        MsgBox(result)

    End Sub
End Class
```

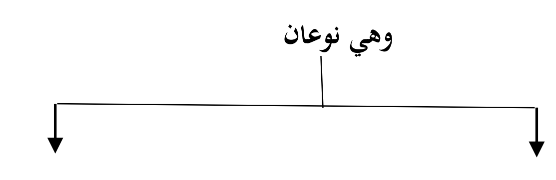
(الشكل٥–٢)يوضح الكود المتعلق بالعملية الحسابية

انواع المتغيرات



متغيرات محلية

وهي المتغيرات التي يتم الاعلان عنها داخل اجراءات الوحدة البرمجية وعمرها البرمجي محصور منذ بدء الاعلان عنها وحتى تنفيذها .



متغيرات محلية ديناميكية

Dynamic Local Variables

```
Dim counter As
Integer
For counter = 1
To 10
Dim y as integer
…
…
Next
```

متغيرات محلية الستاتيكية

Static Local Variables

```
Sub Counter ()
Static counter As
Integer
Counter = counter +
1
…
…
End Sub
Sub Print Data ()
Static is printing
as Boolean
If is Printing Then
Exit Sub
Else
Is Printing = True
End If
…
…
End Sub
```

متغيرات عامة

وهي المتغيرات التي يتم الاعلان عنها على مستوى الوحدة البرمجية ويكون مدى هذه المتغيرات شاملا لجميع اجراءات الوحدة البرمجية ويتم التصريح عن هذه المتغيرات خارج اجراءات الوحدة البرمجية

```
Module Module1
' متغيرات على مستوى الوحدة
Dim x as Integer
Private y as Integer
Sub Main ()
x = 50
…
End Sub
Sub Test ()
y = 10
…
End Sub
End Module
```

الثوابت Constants

الثوابت هي نوع من الحوافظ او الخازنات المؤقتة للبيانات وهي تتشابه الى حد كبير مع المتغيرات **variables** من حيث الوظيفة

لكن هناك فرق مابين المتغيرات والثوابت هو أن الثوابت القيمة المعلنة أو المعرفة هي قيمة ثابتة ولايمكن تغييرها وهذا مثال على ذلك:

```
Const MyConst = 3
```

فوائد تعريف الثوابت :

- يعتبر التعامل معها اسرع في التنفيذ من المتغيرات .
- سهولة اضافة ثوابت جديدة .

انواع البيانات Data Types

بعد أن قمنا بايضاح المتغيرات عزيزي القارئ سنقوم حاليا بايضاح انواع البيانات التي سنستخدمها أو يجب أن تتلائم مع المتغيرات التي قمنا بالتصريح عنها ومن امثلة تلك البيانات (**double**,**string**,**integer**,**times**,**date**....الخ) عزيزي القارئ أن هذه الانواع البيانات التي سنتعامل معها ماهي الا فئات **classes** وتركيبات **Structures** وسنوضح لك في المواضيع بشكل مكثف عن الفئات **classes** وسنوضح لك حاليا عن انواع تلك البيانات :

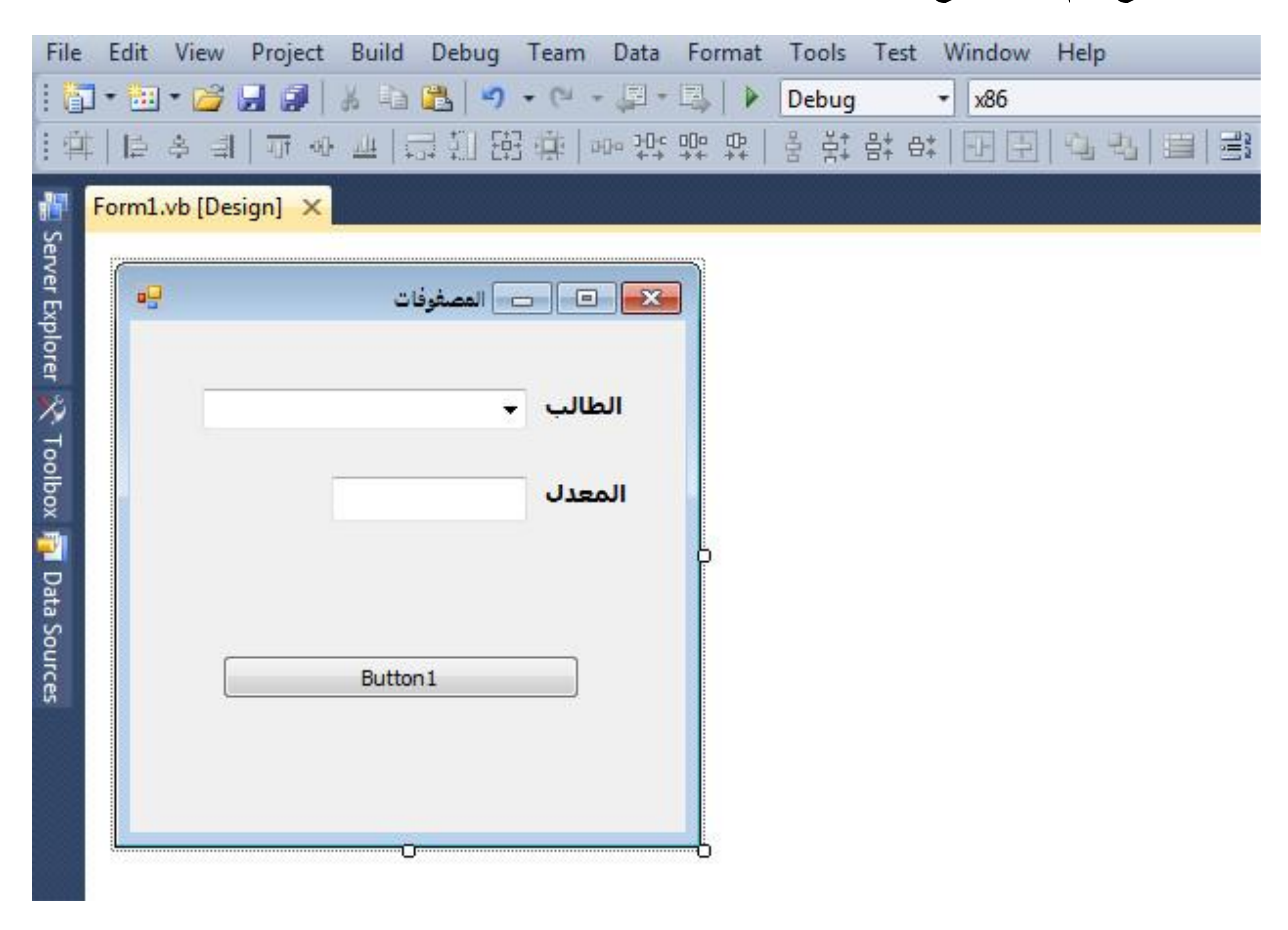
النوع	الحجم	مجال القيمة
Boolean	٢ بايت	**True** (صح) أو **False** (خطأ).
Byte	١ بايت	عدد صحيح من **0** الى **255** .
Char	٢ بايت	حرف واحد من نوع **Unicode**
Date	٨ بايت	وقت من الساعة 0:00:00 الى الساعة 11:59:59 كما يشمل تاريخ من يوم ١ يناير لعام **0001** الى **31** ديسمبر لعام **9999**.
Decimal	١٦ بايت	عدد صحيح من **0** إلى**Decimal** **79,228,162,514,264,337,593,543,950,335**± أو عشري من **0** إلى **7.9228162514264337593543950335**±
Double	٨ بايت	يتضمن اعداد بها كسور.
Integer	٢بايت	يتضمن اعداد صحيحة.
string	١٠+(٢*عددالحروف)بايت	من **0** الى **2** مليار حرف غير مشفرة

المصفوفات Arrays

المصفوفات : هي طريقة لجمع مجموعة من المتغيرات مترابطة بطريقة ما ، أو هو متغير واحد يحتوي على عدة بيانات مترابطة لكل بيان داخلها له رقم خاص به من اجل الوصول اليه ويمكن استخدام **Arrays** مثلا لحفظ درجات امتحان الطلاب بفصل محدد أو بقائمة اسعار ولمعرفة كيفية التعامل مع المصفوفات نوضح لكم ذلك من خلال التمرين التالي :

تمرين :

قم بانشاء برنامج مصفوفات يقوم بعرض اسماء الطلاب وذلك باستخدام المصفوفات وقم بافتراض ان عدد المتغيرات الفرعية للمصفوفة هي **3** متغيرات تتضمن اسماء الطلاب : محمد ، ايهاب ،أحمد واطلب من البرنامج أن يقوم بعرض المتغير الفرعي **array(2)** في رسالة نصية وسنوضح لكم الان البرنامج بعد اضافة الادوات الية كما (بالشكل ٦-٢) :



(الشكل٦-٢) يوضح اضافة ادوات برنامج المصفوفة

والى كتابة الكود المتعلق بالمصفوفة داخل **Button1** كما ستلاحظون في (الشكل ٧-٢)

```
Button1                                   Click
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Dim myarray(2) As String
        myarray(0) = "محمد"
        myarray(1) = "ايهاب"
        myarray(2) = "أحمد"
        MsgBox(myarray(2))

    End Sub
End Class
```

(الشكل ٧-٢) المتعلق باضافة كود المصفوفة

وعند القيام بتنفيذ البرنامج بالضغط على **F5** من اجل مشاهدة الرسالة الناتجة عن تنفيذ المصفوفة انظر (الشكل٨-٢)



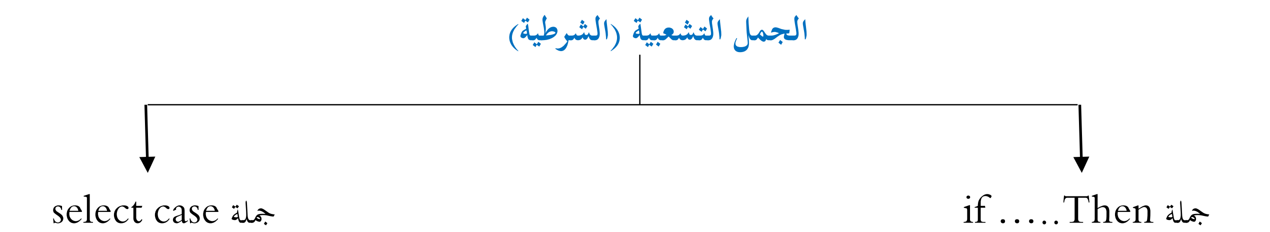
(الشكل٨-٢) يوضح ظهور الرسالة النصية باختيار عنصر المصفوفة (2)**array**

انواع الجمل Statements

الجمل التي يمكن استخدامها هنا لدينا نوعان منها :

- جمل تشعبية (شرطية)
- جمل حلقات تكرارية

وسنوضح لك عزيزي القارئ تفاصيل كل نوع من هذه الجمل بالشرح والامثلة وسنبداء اولا بالجمل التشعبية (الشرطية) :



اولاً جملة if .....Then:

وهي عبارة عن جملة شرطية تستخدم بتحقيق شرط معين داخل جملة محددة وسنوضح لك عزيزي القارئ ذلك بمثال :

تمرين :

وفقاً لمثالنا السابق لدينا نقوم بتعبئة اداة ال **combobox1** باسماء الطلاب : محمد ، ايهاب ، أحمد , ومن ثم لنفترض انه في حال تم اختيار اسم الطالب ايهاب من ال **combobox1** يقوم البرنامج بعرض نتيجة الطالب (المعدل) في مربع نص المعدل واذا لم يختار اي اسم من هؤلاء الطلاب تظهر له رسالة بانه لم يختار شئ ولتحقيق ذلك ولتعريف متغيرات المصفوفة الفرعية المتعلق بالدرجات للطلاب انظر الجدول بالاسفل وهو يضم اسم الطالب والدرجة النهائية أو المعدل :

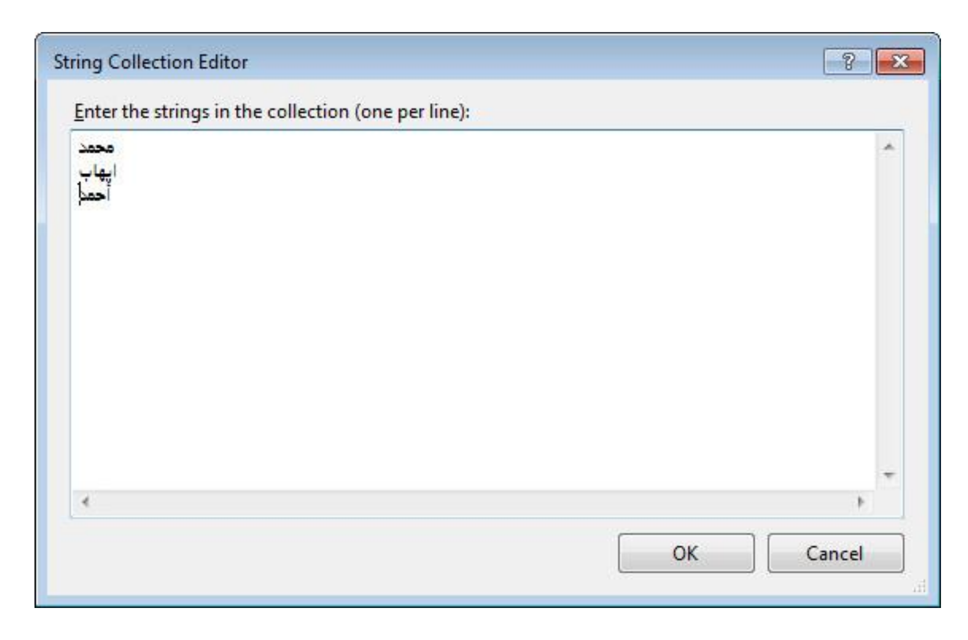
م	اسم الطالب	المعدل	تعريف المتغير الفرعي الخاص بالمتغيرات
١	محمد	%٩٠	dig(0)
٢	ايهاب	%٨٥	dig(1)
٣	أحمد	%٨٠	dig(2)

والان الى شاشة اضافة اسماء الطلاب الى القائمة **combobox1** من خلال نافذة الخصائص ثم نذهب الى **items** العناصر كما ستلاحظون (بالشكل٩-٢)

DisplayMemb	(none)
Items	(Collection)
Tag	

(الشكل٩-٢)

ومن ثم اضافة الاسماء في نافذة العناصر كما (بالشكل١٠-٢):



(الشكل١٠-٢) المتعلق باضافة الاسماء الى قائمة عناصر ال **combo box**

ومن ثم تعديل عنوان **Button1** الى النتيجة ، بعد ذلك نقوم بالذهاب الى كتابة كود المصفوفة والجملة التشعبية **if...then** المتعلق بعرض نتائج الطلاب في مربع النص ونقوم بكتابة الكود في **Button1** كما هو موضح (بالشكل١١-٢) :

```
Button1                                   Click

        Dim dig(2) As Integer
        dig(0) = 90
        dig(1) = 85
        dig(2) = 80
        If ComboBox1.Text = "" Then

            MsgBox("لم تختار شئ ؟")
        Else

            If ComboBox1.Text = "ايهاب" Then
                TextBox1.Text = dig(1)

            End If
            If ComboBox1.Text = "محمد" Then
                TextBox1.Text = dig(0)

            End If
            If ComboBox1.Text = "أحمد" Then
                TextBox1.Text = dig(2)

            End If
```

(الشكل١١-٢) يوضح اضافة الكود المتعلق ببرنامج نتائج الطلاب

اعداد / محمد حازم الشميري

ملاحظة :

كما تلاحظ عزيزي القارئ انه تم تكوين مصفوفة تتعلق بدرجات الطلاب تتكون من ٣ عناصر فرعية بمعنى dig(2) وبعد تكوين المصفوفة قمنا بكتابة الجملة التشعبية **if...then** وهي تتضمن في سطرها الاول

```
If ComboBox1.Text = "" Then
```

بمعنى انه لو انه لايوجد أي عناصر تم اختيارها من القائمة يتم تنفيذ الاتي :

عرض رسالة نصية تخبرك بانك لم تختار شئ

```
MsgBox ("لم تختار شئ ؟")
```

ثم نقوم بعد ذلك بفصل الشرط الاول عن الي بعده من خلال استخدام كلمة

```
Else
```

بعد الكود السابق لنقوم بعد ذلك بكتابة الشرط التالي لو ان العنصر المختار من القائمة هو "ايهاب" قم بعرض النتيجة الموجودة في المتغير الفرعي في مربع النص كما هو موضح بالكود

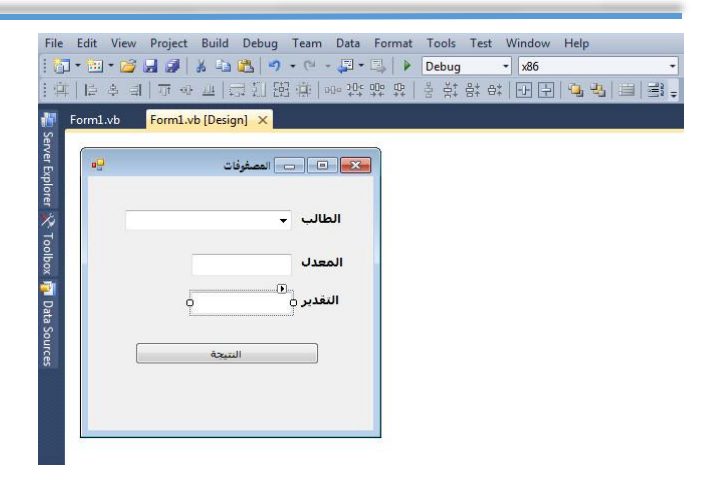
```
If ComboBox1.Text = "ايهاب" Then
TextBox1.Text = dig (1)

End If
```

وكذلك مع بقية العناصر يتم التعامل بنفس الطريقة.

ثانياً جملة Select case :

وهي عبارة عن احدى انواع الجمل التشعبية ويتميز هذا النوع من الجمل بالسهولة والسلاسة في تحقيق محتواه وسنوضح لكم استخدام الجملة التشعبية **Select Case** من خلال الاستعانة بمثالنا السابق واضافة اليه انه يقوم البرنامج بعرض نوع المستوى الذي حققه الطالب من خلال المعدل الحاصل عليه فلو انه حصل على معد ٩٠% بمعنى أن تقديره امتياز واذا كان تقديره ٨٥% هذا يعني حصل على تقدير جيد جداً ولتنفيذ ذلك نقوم باضافة مربع نص اخر لاظهار التقديرات **Textbox2** واضافة **label** عنوان لمربع النص كماهو واضح امامكم (بالشكل ١٢-٢) :



(الشكل١٢–٢) المتعلق باضافة مربع نص التقدير وعنوان **label**

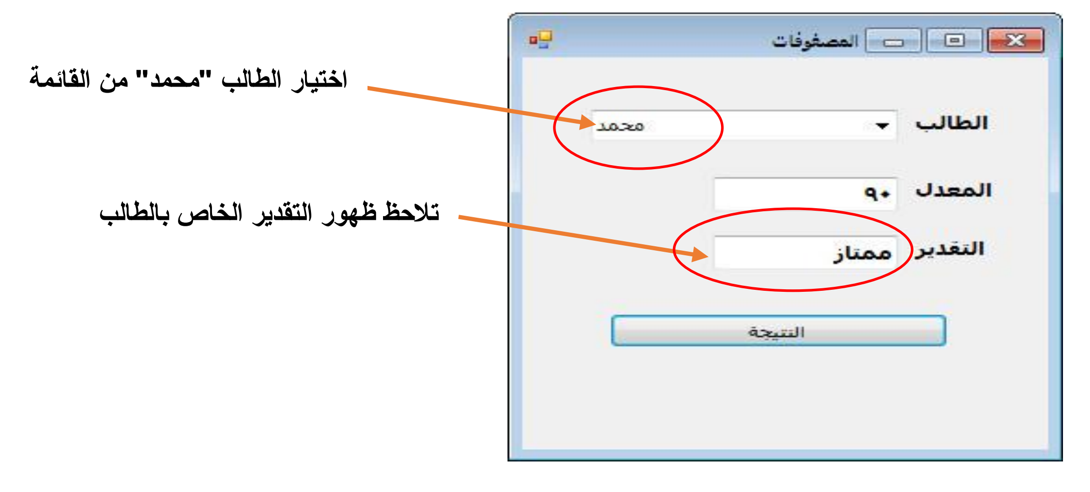
وبعد ذلك نقوم بالانتقال الى ادخال الكود المتعلق بتنفيذ عرض التقديرات ندخل الكود في **Button1** كما تلاحظ في (الشكل١٣–٢) :

```
        Select Case TextBox1.Text.ToString
            Case Is = 90
                TextBox2.Text = "ممتاز"
            Case Is = 85
                TextBox2.Text = "جيدجداً"
            Case Is = 80
                TextBox2.Text = "جيداً"

        End Select
    End Sub
End Class
```

(الشكل١٣–٢) المتعلق بادخال كود جملة **select case**

وبعد أن تم ادخال كود البرنامج قم بتشغيل البرنامج وقم باختيار الطالب محمد على سبيل المثال من القائمة ستلاحظ انه تم اظهار تقدير الطالب محمد في مربع نص التقديرات كما تلاحظ في (الشكل ١٤–٢) :



(الشكل ١٤–٢) الخاص باظهار مربع التقديرات للطلاب

الحلقات التكرارية

• الحلقة التكرارية For/Each

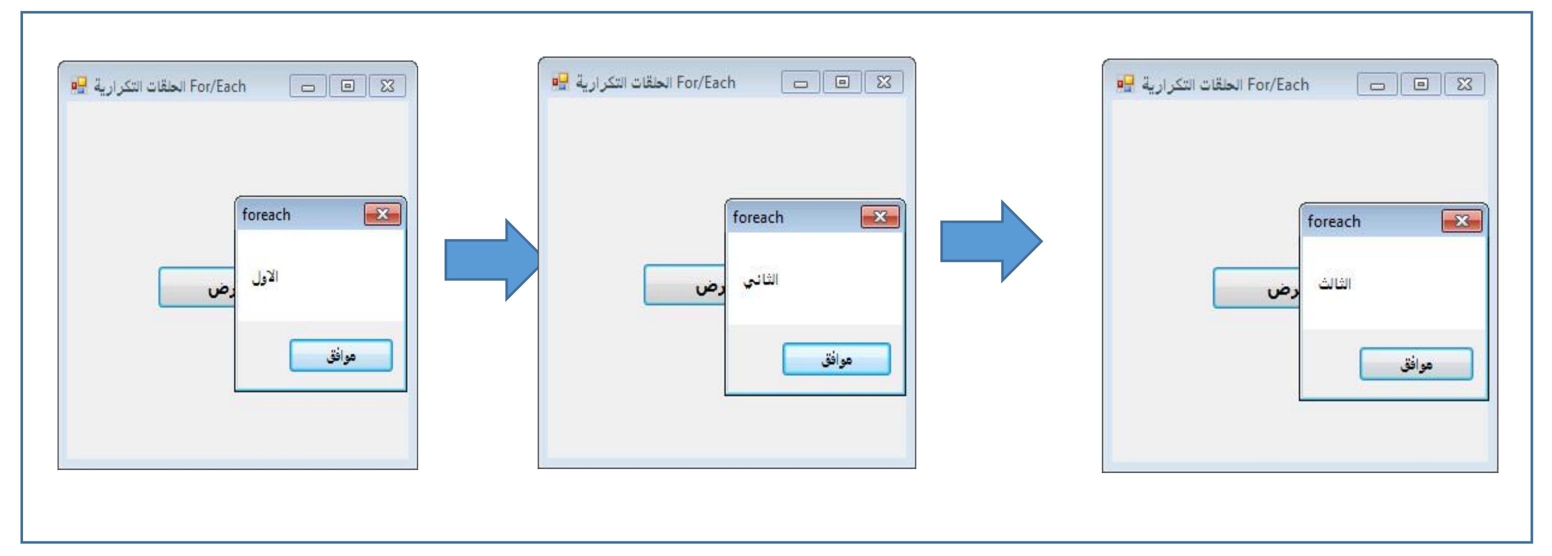
الحلقات التكرارية هي عبارة عن جمل تتيح لك القيام بعملية التكرار داخل جزء من الكود أو ممكن انها تمر على متغير فرعي متعدد المحتوى كما في المثال التالي :

قم بانشاء برنامج جديد واضف اليه **Button1** بحيث يقوم البرنامج بعرض ناتج متغيرات نصية لجملة تكرارية من نوع **For/ Each** سنوضح لكم اضافة **Button1** في النموذج وذلك في (الشكل١٥-٢) :

```
Dim myarray() As String = {"الاول", "الثاني", "الثالث"}
'في هذ المتغير قمنا بتعريفه الى ٣ جمل او متغيرات فرعية وهي الاول ,الثاني,الثالث
Dim level As String
' وهنا قمنا باعلان متغير من نوع string
For Each level In myarray
    MsgBox(level)
    'في متغير الحلقة التكرارية myarray هنا قمنا بتمرير المتغير level
    ' في رسالة نصية الى الوصول باخرمتغير فرعي وهو الثالث button الفرعية كل مره يتم النقر على array وسيتم عرض متغيرات
Next
```

(الشكل١٥-٢) يوضح طريقة اضافة كود الحلقة التكرارية داخل **button1**

قم بتشغيل البرنامج ب **F5** ومن ثم قم بالنقر على **Button1** انظر(الشكل١٦-٢) وستلاحظ فيه ظهور الرسالة بشكل تكراري كلما مرة تقوم بالنقر على موافق داخل الرسالة التي ستظهر لك :



(الشكل١٦-٢) يوضح ظهور الرسالة بشكل تكراري بعدد المتغيرات الفرعية داخل المصفوفة .

• الحلقة التكرارية For/Next

تستخدم هذه الجملة لتكرار جزء من الكود عدد من المرات يتم تحديدها مسبقا سيتم شرحها من خلال المثال التالي :

تمرين:

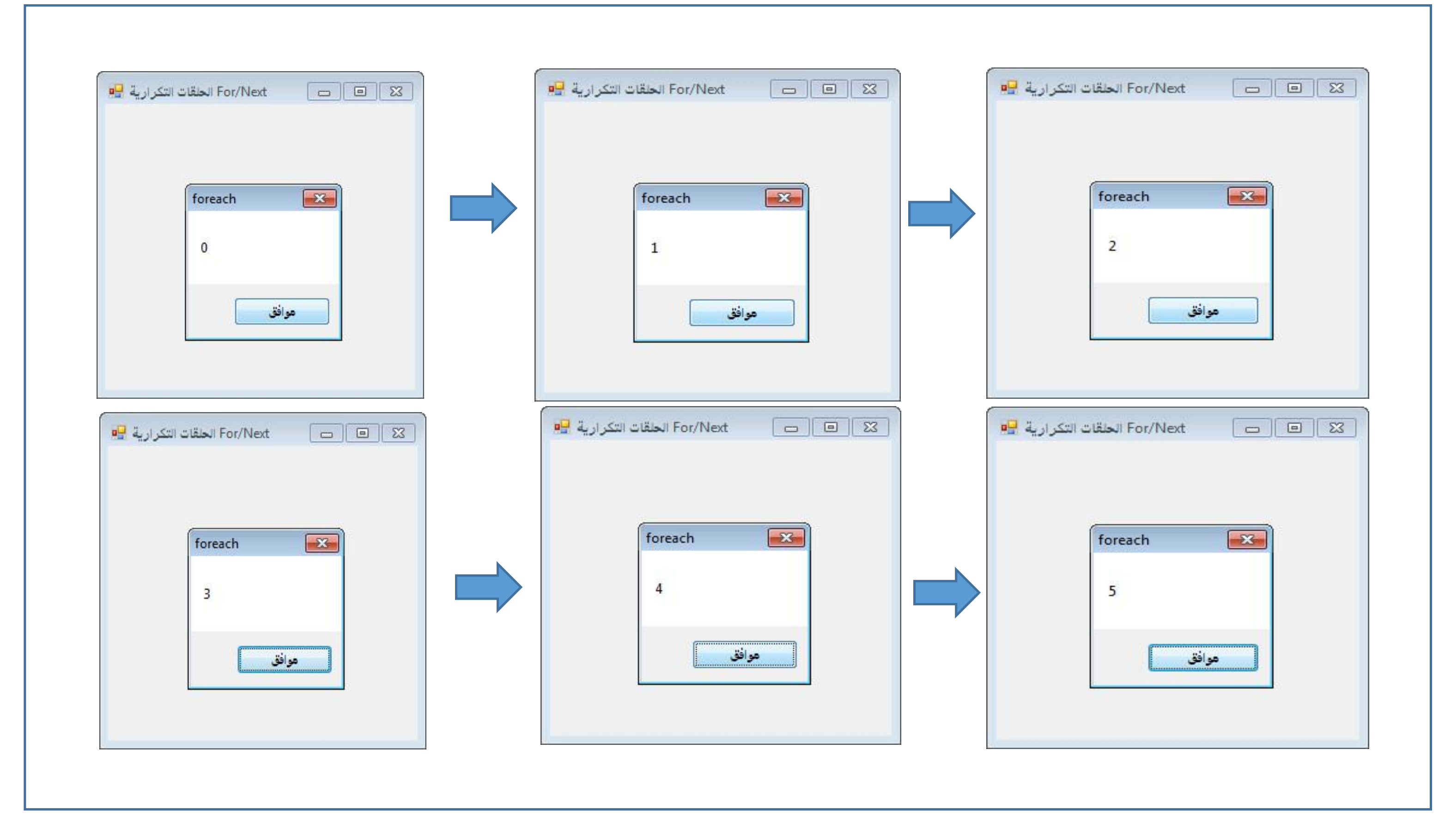
من خلال الاستعانة بالمثال السابق نقوم الان باضافة كود الحلقة التكرارية For/Next فلنقوم باعلان عن متغير رقمي يتكرر من 0 الى 5 ويتم عرض مرات التكرار في رسالة نصية عدد 5 مرات عند النقر على Button1

نقوم باضافة الكود التالي في Button1 :

```
Dim I as Integer
For I = 0 To 5
MsgBox (i.ToString)

Next
```

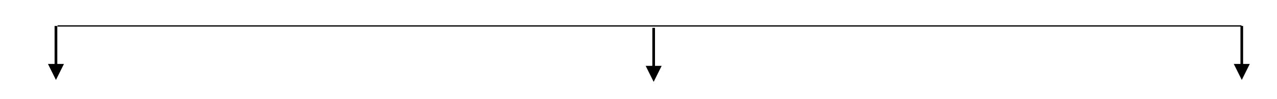
ستلاحظ عزيزي القارئ هذه المره انه تم اعلان متغير من نوع **integer** وفي السطر الثاني من الكود يتم كتابة الجملة التكرارية على ان يحمل المتغير **I** عدد مرات التكرار من **0** الى **5** وفي السطر الثالث يتم عرض مرات التكرار في رسالة نصية كما تلاحظ في (الشكل١٧-٢) :



(الشكل١٧-٢) يوضح تطبيق كود جملة **For/Next** وظهور الرسالة النصية عدد المرات التي اعلن فيها المتغير **I**

مقدمة عن تصحيح الاخطاء

من المالوف عزيزي القارئ انه عند الانتهاء من بناء مشروعك الخاص في **Visual Basic.Net** فانه لابد من تواجد اخطاء وقد تكون هذه الاخطاء اما ان تم اكتشافها من قبل لغة البرمجة اثناء القيام بعملية التنفيذ أو الاختبار أو اخطاء لاتستطيع اكتشافها لغة البرمجـــة فلابد من تواجد مثل هكذا اخطاء مهما كانت المهارة والاحترافية العالية من قبل المبرمج وسنذكر لك عزيزي القارئ حاليا ٣ انواع من تلك الاخطاء :



اخطاء املائية

وهي تلك الاخطاء الناجمة عن خلل املائي في جملة الكود من نقص للاحرف أو أن تكون الكلمات المستخدمة غير صحيحة ولاشان لها بالبرمجة وسنذكر مثالين لمثل هكذا اخطاء :

المثال الاول :

Dim s a string = "Mohammed"

نجد أن الخطاء هنا هو في جملة الاعلان كلمة as التي يجب تواجدها بعد اسم المتغير وقد تم ذكرها ب "a"

المثال الثاني :

```
Dim a as string
If a > 50 then
A= a+50
```

لاحظ الخطأ هو عدم ذكر end if التي تختتم جملة if الشرطية

اخطاء تشغيلية

وهي تلك الاخطاء الناجمة عن اخطاء يتم اكتشافها اثناء التشغيل واحيانا بعد الانتهاء من انشاء المشروع يتم اكتشافها ومثل هذه الاخطاء هو محاولة الاتصال بقاعدة البيانات مع عدم صحة مسار قاعدة البيانات أو وجود خطاء في بناء جمل قواعد البيانات ...الخ ويمكن معالجة مثل هكذا من الاخطاء باستخدام عبارات أو جمل برمجية ومنها اضافة جملة on error resume next

تضاف هذه الجملة في بداية الكود البرمجي

وكذلك استخدام جملة Try catch

وتختتم بجملة end try

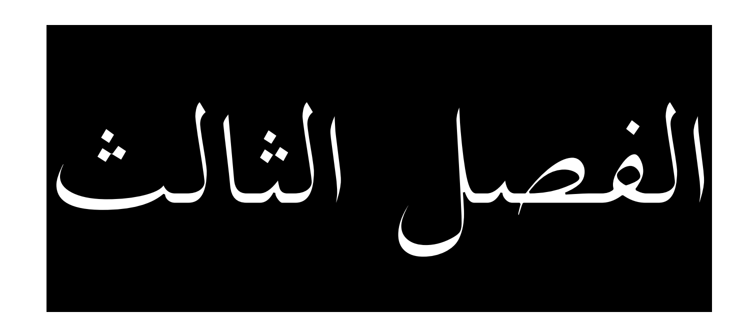
ويوضع الكود البرمجي فيما بين الجملتين

اخطاء منطقية

وهي تلك الاخطاء التي يصعب اكتشافها عندما تكون كفاءة النظام في اقل مايمكن

مثال على ذلك :

```
Dim a As Integer
a = 5
Do While a < 5
a = a - 5
Loop
```



التعامل مع صندوق الادوات Tool box

سنتطرق في هذا الفصل الى اسس التعامل مع صندوق الادوات Tool box موضحين كيفية التعامل مع خصائص واحداث تلك الادوات من

خلال امثلة وتمارين للتوضيح بذلك.

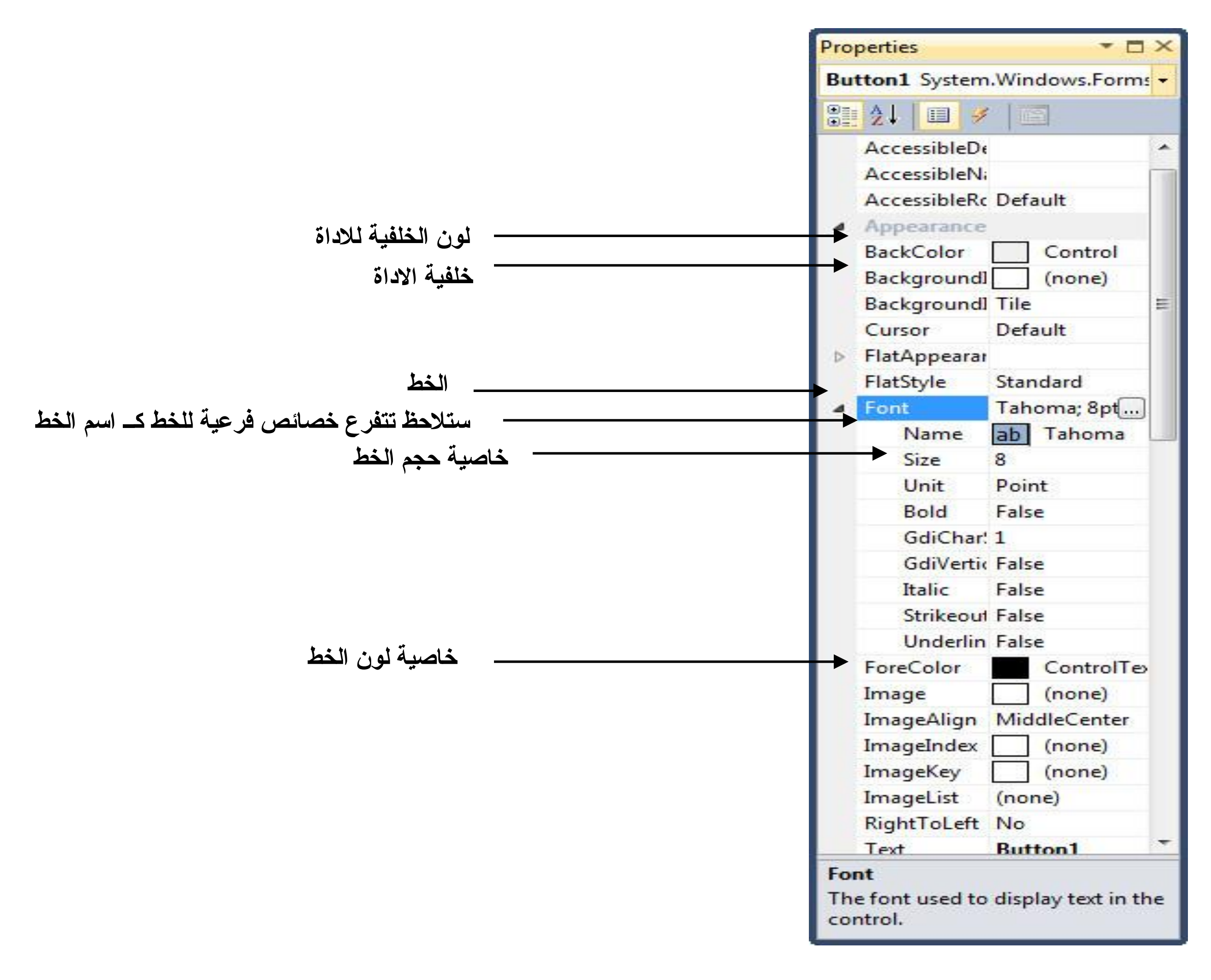
- **مدخلك للتعامل مع صندوق الادوات Tool box**
- **الخصائص والاحداث**

مدخلك للتعامل مع صندوق الادوات Tool box

عزيزي القارئ سبق وأن تحدثنا بشكل مختصر عن صندوق الادوات Tool box في الفصل الاول و او ضحنا اليك طريقة اضافة عناصر جديدة في صندوق الادوات Tool box وكيفية اضافة الادوات أو طرق اضافة الادوات الى النموذج سنقوم بشرح تفاصيل صندوق الادوات Tool box كل اداة على حدة وسنوضح خصائص واحداث تلك الادوات والاساليب موضحين كل ذلك بمجموعة من الامثلة والتمارين الشيقة .

الخصائص properties

عزيزي القارئ بعد أن عرفت نبذة مختصرة عن نافذة الخصائص في الفصل الاول ستتعرف في هذا الفصل تفاصيل تلك النافذة وطرق التعامل بشكل اكثر موضحين ذلك بمجموعة من الامثلة والتمارين وقبل أن نتطرق الى ذلك سنوضح لكم نافذة الخصائص properties انظر (الشكل ١-٣) :



(الشكل١-٣) لتوضيح نافذة الخصائص

كيفية تعيين خصائص الادوات ؟

كما تحدثنا عزيزي القارئ في الفصل الاول توجد طريقتين لتعيين الخصائص :

الاولى : تعيين الخصائص بشكل يدوي ، أي بمعنى أن المبرمج يقوم بتعيين وتعديل خصائص الادوات المضافة الى النموذج بشكل يدوي .

الثانية : عن طريق الكود البرمجي وباستخدام الاحداث **Events** وكما ذكرنا أن هذه الطريقة تعتبر طريقة تتسم بالمرونة أي بمعنى أن المبرمج يستطيع تعيين وتعديل خصائص الادوات المضافة الى النموذج من خلال الاكواد المتعلقة أو المنسجمة مع الخصائص كذلك يستطيع المبرمج أن يجعل امكانية تعيين الخصائص بظل هذه الطريقة متاحة للمسخدم من خلال انشاء نافذة للخصائص تقوم بتعيين وحفظ التعيينات بحيث يستطيع المستخدم العودة لها مرة اخرى .

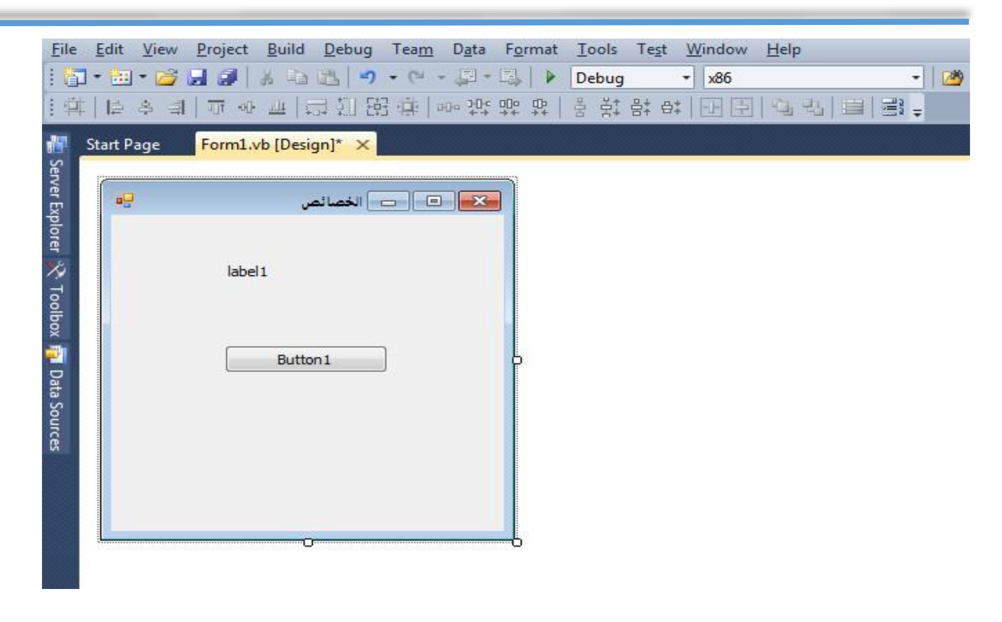
عزيزي القارئ سنبداء اولاً بشرح الطريقة الاولى من خلال تمرين بسيط :

تمرين :

قم باضافة الادوات المذكورة بالجدول الى نموذجك الحالي وقم بتغيير الخصائص الواردة في الجدول يدوياً :

اسم الأداة	الخاصية	القيمة
Button1	**Name** **Text**	**Btn1** تغيير الخصائص
Label1	**Name** **Text** **Font. Size** **Font. Bold** **Back color** **Fore Color**	**Lbl1** بسم الله الرحمن الرحيم **18** **True** **Black** **White**

انظر الجدول وهو يوضح اشكال توضيحية لاضافة تلك الادوات الى النموذج ويقوم بتغيير تلك الخصائص يدوياً كما انك ستلاحظ في (الشكل ٢-٣) اضافة الادوات الى النموذج اولاً :



(الشكل٢-٣) وهو يوضح اضافة الادوات المضافة للنموذج

الاداة	الشرح التفصيلي
Button1 -١	- تغيير خاصية Name وذلك من خلال الذهاب الى نافذة الخصائص ومن ثم تغيير اسم الزر الى btn1 كما هو في (٣-٣) : تغيير اسم الاداة → (Name) btn1 (الشكل٣-٣) يوضح تغيير خاصية الاسم Name - تغيير خاصية النص Text الى "تغيير الخصائص" كمافي (الشكل٤-٣) : تغيير نص الاداة → Text تغيير الخصائص (الشكل٤-٣) يوضح تغيير الخاصية Text
Label1 -٢	- تغيير خاصية الاسم Name للاداة الى lbl1 كما في (الشكل ٥-٣):

اعداد / محمد حازم الشميري

(الشكل٥-٣) يوضح تغيير الخاصية Name الى labl1

- تغيير خاصية النص Text الى "بسم الله الرحمن الرحيم" وذلك كما (بالشكل٦-٣) :

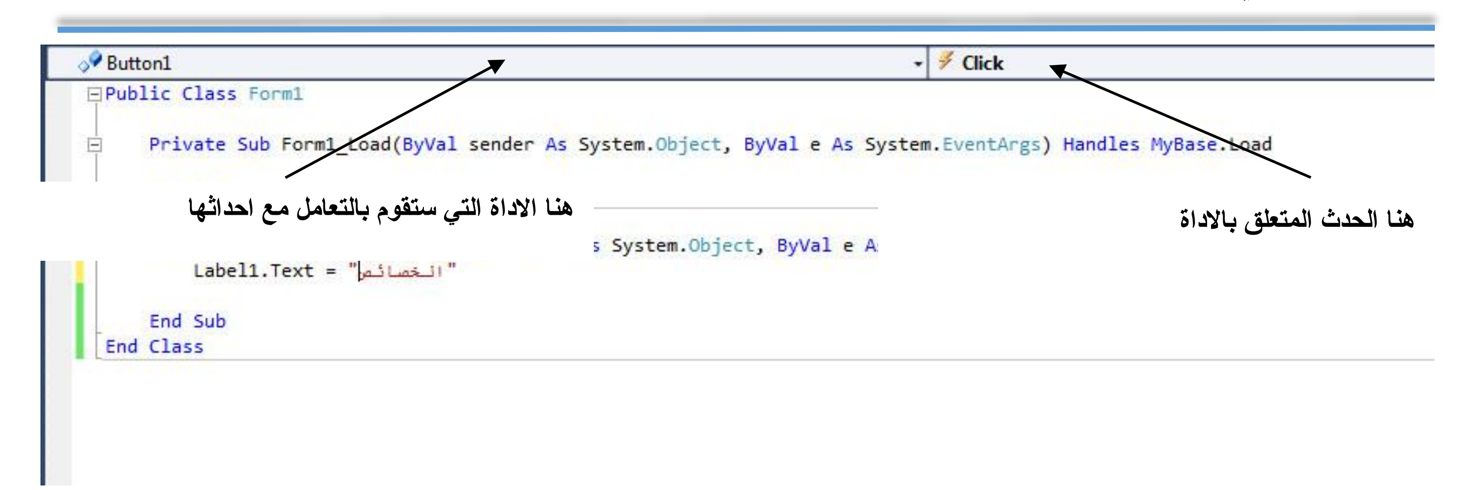
(الشكل٦-٣) يوضح تغيير الخاصية Text الى "بسم الله الرحمن الرحيم"

- تغيير خواص الخط (الحجم ، عريض) كذلك تغيير خاصية Back color لون الخلفية الى black وتغيير خاصية لون الخط الى white انظر (الشكل٧-٣) ولاحظ تغيير الخصائص :

(الشكل٧-٣) وهو يوضح تغيير خاصية Fore ,Back color ,Font color .

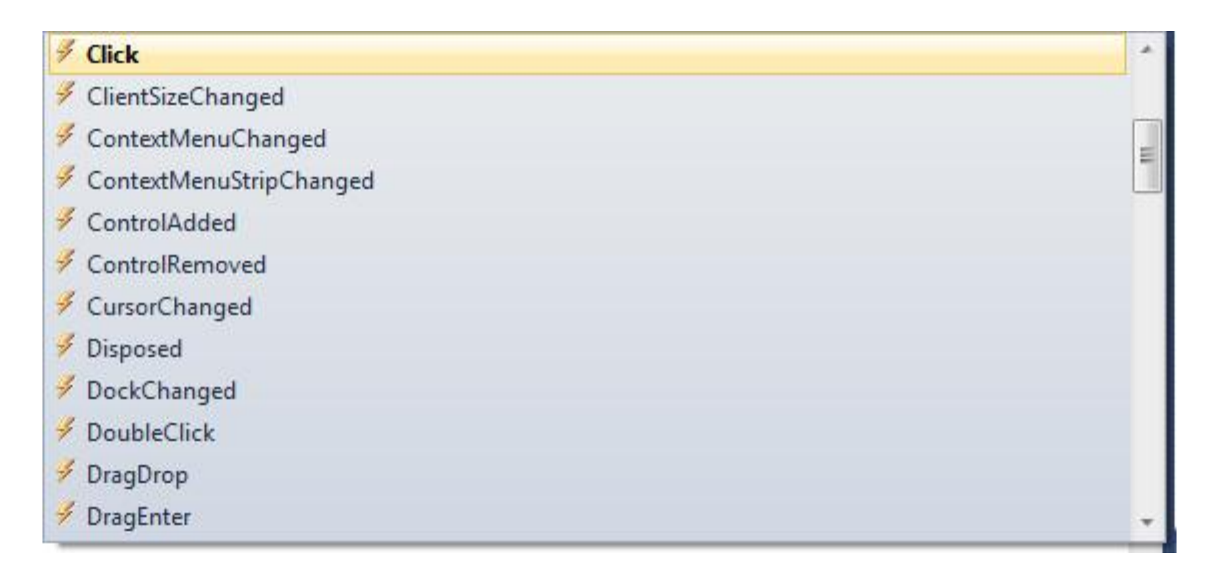
الاحداث Events

مفهوم الاحداث Events هي المكان المناسب والذي توضع في الاوامر للتنفيذ بزمن محدد ، فالاحداث تعتبر هي الطريقة الثانية لتعيين خصائص الادوات فالاحداث تجدها في نافذة تحرير الكود كما في (الشكل٨-٣) :



(الشكل٨–٣) يوضح مكان الاحداث في نافذة تحرير الكود

يمكنك عزيزي القارئ التعامل مع مختلف احداث الكود المتواجدة في القائمة اليمنى للنافذة بكل سهولة سنقوم بعرض مجموعة من الاحداث ونوضح كل حدث مع مثال بسيط لذلك انظر (الشكل٩–٣) وهو يوضح مجموعة من الاحداث في قائمة الاحداث:

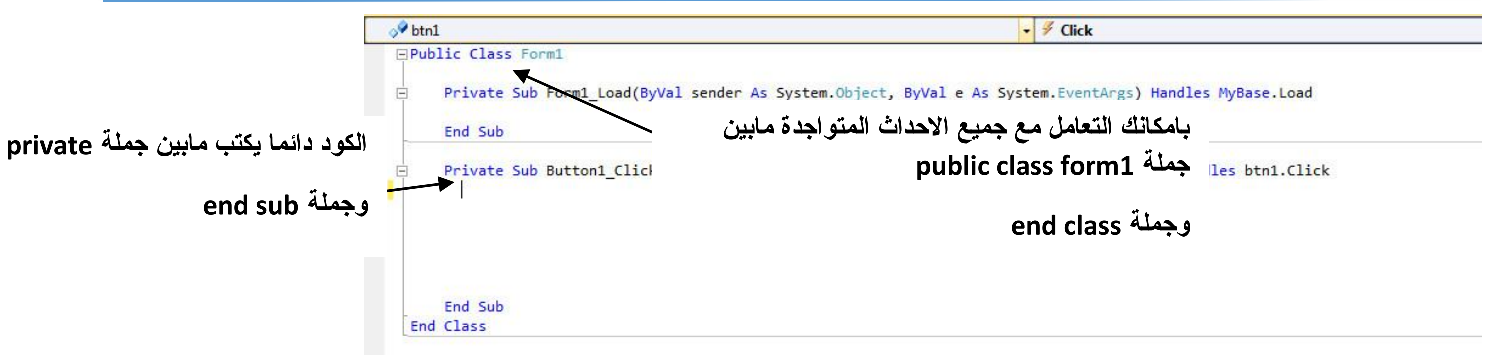


(الشكل٩–٣) يوضح قائمة تضم مجموعة من الاحداث في قائمة الاحداث

ستلاحظ عزيزي القارئ ان كل حدث يبداء بجملة اعلان عن الاجراء المتعلق بالحدث ويختتم الحدث بجملة **End Sub**

على سبيل المثال لناخذ الحدث **click** ستجد أن يتكون من جملة الاعلان عن الاداة **button1** وينتهي بجملة **end sub**

كما في (الشكل١٠–٣):



(الشكل ١٠-٣) يوضح تركيب الحدث Click

عزيزي القارئ سنقوم بايضاح مجموعة من الوظائف لمجموعة من الاحداث المتعلقة ب **Button1** على سبيل المثال من خلال الجدول بالاسفل :

الحدث	الوظيفة
– **Click** – **Double click**	الحدث **click** وظيفته تنفيذ كود محدد عند القيام بعملية النقر على **Button1** الحدث **double click** يقوم بتنفيذ كود محدد عند القيام بعملية نقر مزدوج على **Button1** .

كيفية تعيين الخصائص من خلال الاحداث :

يتم تعيين الخصائص من خلال ادخال كود برمجي محدد في حدث ما للتحكم في خاصية معينة من خواص الادوات المضافة الى النموذج وللتوضيح اكثر من خلال تمرين يوضح كيفية تقوم بتغيير خواص **label** في نموذجك من خلال استخدام الحدث **click** للزر **Button1** .

تمرين :

من خلال الاستعانة بالتمرين السابق لدينا قم بتغيير الخصائص للاداة **Button1** و **label1** كما هو مطلوب منك بالجدول اعلاه .

من خلال الاستعانة (بالشكل٢-٣) وهو يوضح اضافة الادوات الى النموذج نقوم الان بالدخول الى نافذة تحرير الاكواد لتنفيذ الكود المتعلق بتغيير الخصائص من خلال الحدث **click** لـ **button1** وذلك من خلال النقر المزدوج على **button1** ومن ثم نقوم بادخال كود تغيير الخصائص :

Code	Visual Basic.Net

```
Private Sub Button1_Click (By Val sender As System. Object, By Val e As System.EventArgs) Handles
btn1.Click
Button1.Text = "تغييرالخصائص"
Label1.Text = "بسم الله الرحمن الرحيم "
Label1.Font = New Drawing. Font("Tahoma", 18, FontStyle.Bold, GraphicsUnit.Pixel)
Label1.BackColor = Color. Black
Label1.ForeColor = Color. White
Label1.Name = "lbl1"
```

اعداد / محمد حازم الشميري

```
Button1.Name = "btn1"
End Sub
```

وهو كما هو ظاهر في (الشكل١١-٣) الذي يوضح نتيجة الكود السابق في تغيير خصائص الاداة Button1 & label1



(الشكل١١-٣) يوضح تغيير خصائص الادوات من خلال الحدث click

ملاحظة :

- في السطر الاول من الكود السابق قمنا بتغيير خاصية النص للاداة Button1 الى "تغييرالخصائص"

```
Button1.Text = "تغييرالخصائص"
```

- في السطر الثاني قمنا بتغيير خاصية النص للاداة labl1 الى "بسم الله الرحمن الرحيم"

```
Label1.Text = "بسم الله الرحمن الرحيم"
```

- في السطر الثالث قمنا بتغيير خواص الخط للاداة label1 { اسم الخط ، حجم الخط،عريض}

```
Label1.Font = New Drawing. Font ("Tahoma", 18, FontStyle.Bold,
GraphicsUnit.Pixel)
```

- في السطر الرابع قمنا بتغيير خاصية لون خلفية الخط الى اللون الاسود

```
Label1.BackColor = Color. Black
```

- في السطر الخامس قمنا بتغيير خاصية لون الخط للاداة label1 الى الابيض

```
Label1.ForeColor = Color. White
```

- في السطر السادس قمنا بتغيير خاصية الاسم للاداة label1 الى lbl1 ستلاحظ عزيزي القارئ اننا قمنا بتاجيل تاخير تغيير الاسم الى السطور الاخيرة لماذا؟
حتى نتيح تنفيذ الاوامر السابق على الاسم السابق label1 ونتجنب حدوث اخطاء .

```
Label1.Name = "lbl1"
```

- كذلك هنا قمنا بالسطر الاخير قمنا بتغيير خاصية الاسم للاداة button1 الى btn1 ستلاحظ اننا قمنا كذلك
- بتاخير اعادة تسمية الاداة لكي تجنب حدوث الاخطاء وجعلناها بالاخير.

```
Button1.Name = "btn1"
```

• اداة العنوان

A Label

استخدامها	تستخدم لتوصيف ادوات اخرى في الغالب وتصبح عنوانا لتلك الاداة .
مثال	قم بانشاء برنامج يقوم بادخال بيانات تسجيل درجات الطلاب في مربع نص Textbox واضف اليها مربع نص لتوصيفها باسم "درجات الطالب" كما (بالشكل١٢-٣) : خصائص اداة العنوان درجات الطالب (الشكل ١٢-٣) يوضح اضافة اداة العنوان

خصائص اداة العنوان label

١- خاصية الاسم Name

من خلال هذه الخاصية تستطيع أن تتحكم في تغيير اسم الاداة .

الخاصية	القيمة	الشرح
- Back color		• تستخدم هذه الخاصية للقيام بتغير خلفية النص الموجود في اداة label1 ولتغيير خاصية اللون ماعليك الا أن تذهب الى نافذة الخصائص ثم الخاصية Back color وتقوم من خلالها بتغيير اللون كما في (الشكل ١٢-٣) : Appearance BackColor Control BorderStyle None Cursor Default (الشكل١٢-٣) يوضح تغييرخاصية Back color

الخاصية	القيمة	الوصف
– Border Style		• خاصية الحد تستخدم هذه الخاصية للقيام بالتمييز مابين ادوات العنوان label في حال كان لديك اكثر من اداة مختلفة الوظيفة مثلا لو كان عندك برنامج يتكون من عدة label على شكل مجموعات Group مثلا مجموعة متعلقة ببيانات الطلاب ومجموعة متعلقة ببيانات المدرسين فانت لازم تفرق وتميز بين المجموعتين من خلال هذه الخاصية ويمكنك التحكم بهذه الخاصية باستخدام الكود التالي : Label1.BorderStyle=BorderStyle.Fixed3d وبنشرح لكم كل قيمة ومامعناها والغرض منها
	None	– معناها أن الخاصية Border Style لم يطرئ عليها أي تغيير ولم تضع لها حد وتستخدم لالغاء الحد.
	Fixed Single	– معناه أن الخاصية Border Style قمنا بوضع حد لها من نوع حد خطي كما في (الشكل ١٣-٣) :  (الشكل ١٣-٣)يوضح تعيين القيمة Fixed Single
	Fixed3d	– تقوم هذه القيمة بتغيير شكل الحدود الخارجي لاداة العنوان الى ثلاث الابعاد انظر (الشكل١٤-٣) : (الشكل ١٤-٣) يوضح تعيين القيمة Fixed3d
– Cursor		• تستخدم هذه الخاصية لتغيير شكل مؤشر الفارة عند المرور فوق الاداة label وهناك قائمة تتضمن عدة اشكال من المؤشرات يمكنك تجربتها .

الخاصية	الخاصية الفرعية	القيمة	الشرح
Font	Font Name		تمكنك تحديد اسم الخط الذي تريده.
	Font Bold	– True – False	الخط سميك الخط غير سميك
	Font Size	– رقم صحيح	تمكنك من تحديد حجم الخط .
	Font Strikeout	– True – False	معناه تم وضع سطر اسفل الخط. لم تضع سطر اسفل الخط.
	Font Italic	– True – False	– معناه أن الخط بيكون مائل . – معناه ان الخط غير مائل.

الخاصية	القيمة	الشرح
– Fore color		• تمكنك هذه الخاصية من القيام بتغيير لون الخط للاداة label يمكنك مراجعة تعيين الخاصية من (الشكل السابق ٧–٣) (والشكل ١١–٣)
– Right To Left	– Yes – No	• تستطيع عزيزي القارئ من خلال هذه الخاصية التحكم في عرض نص الخط الموجود في الاداة label1 اثناء التصميم واثناء التنفيذ كذلك . – عرض نص الخط من اليمين الى الشمال . – عرض نص الخط من الشمال الى اليمين .
– Text		• مكان كتابة النص المراد ظهوره في اداة العنوان label1 يمكنك مراجعة ذلك من خلال (الاشكال٦–٣) (١١–٣).
– Text align – Enabled	– Right – Lift – Center	• تمكنك هذه الخاصية من القيام بتحديد اتجاه النص المتواجد في اداة العنوان label1 . – تنسيق النص الى اليمين. – تنسيق النص الى الشمال. – تنسيق النص بالوسط . • تمكنك هذه الخاصية من القيام بتمكين اداة العنوان اثناء التنفيذ.

	- True	- معناه أن اداة العنوان مفعلة . انظر (الشكل١٥-٣) (الشكل ١٥-٣) label1 مفعلة
	- false	- معناه أن اداة العنوان غير مفعلة. انظر (الشكل١٦-٣) (الشكل١٦-٣) label1 غير مفعلة.
- Name		• يمكنك استخدام الخاصية للقيام بعملية اعادة تسمية الاداة label من مثالنا السابق للقيام بعملية اعادة التسمية نقر على الاداة label1 ومن ثم نذهب الى نافذة الخصائص properties ومن ثم نقوم بتغيير الخاصية Name مثلا الى lbl1 كما في (الشكل ١٧-٣) : (الشكل١٧-٣) يوضح تغيير خاصية Name
- Locked		• تستطيع من خلال هذه الخاصية التحكم في امكانية الاتاحة لتفعيل بعض الخواص الاخرى لاداة العنوان label1 .

	– True	– بهذه الحالة فان اداة العنوان مقفلة لن تستطيع التحكم بها من تغيير حجم او تفعيل اي خاصية الا بعد الغاء القفل لها .
	– False	– بهذه الحالة فانك تستطيع تغيير العديد من الخصائص للاداة label1 من حجم وموقع..الخ
– Dock		• تستطيع من خلال هذه الخاصية تحديد مكان ظهور اداة العنوان label1 اثناء التصميم واثناء التنفيذ .
	– Right	– اذا اردت أن تملئ الاداة الجهة اليمنى من النموذج وتتخذ شكل عمودي انظر (الشكل١٨–٣) : (الشكل ١٨–٣) قمنا بملئ الاداة label1 في الجهة اليمنى
	– Left	– يمكنك من خلال القيمة Left أن تملئ الاداة بالجهة الايسرى انظر (الشكل١٩–٣): (الشكل١٩–٣) قمنا بملئ الاداة label1 في الجهة اليسرى • تمكنك هذه الخاصية بالتحكم في حجم اداة العنوان وظهورها بالحجم المطلوب اثناء التصميم واثناء التنفيذ كذلك .

الخاصية	القيم	الوصف
	– Top – Bottom – Fill – None	– نستخدمها في حالة اردنا أن نملئ الاداة بالجزء العلوي للنموذج . – نستخدمها بحال اردنا أن نملئ الاداة بالجزئ السفلي للنموذج . – بحال اردنا أن تملئ الاداة النافذة بكلها . – اذا لم ترغب باستخدام هذه الخاصية .
– Size	– Width – Height	• تستخدم هذه الخاصية للتحكم في حجم ظهور الاداة اثناء التصميم وعند التنفيذ . – تستخدم لتحديد عرض الاداة . – تستخدم لتحديد طول الاداة .
– Visible	– True – False	• تستخدم هذه الخاصية للتحكم في عملية ظهور واخفاء اداة العنوان label1 اثناء التنفيذ . – تستخدم هذه القيمة لاظهار اداة label1 اثناء التنفيذ. – تستخدم هذه القيمة لاخفاء اداة label1 اثناء التنفيذ.
– Location	– X – Y	• تستطيع من خلال هذه الخاصية بالتحكم في ظهور اداة العنوان label1 باستخدام المحورين x ,y . – المستوى x وهو يمثل المستوى الافقي ويتحكم في ظهور الاداة label1 بمكان محدد على المستوى الافقي – المستوى y وهو يمثل المستوى الراسي ويتحكم في ظهور الاداة label1 بمكان محدد على المستوى الراسي .

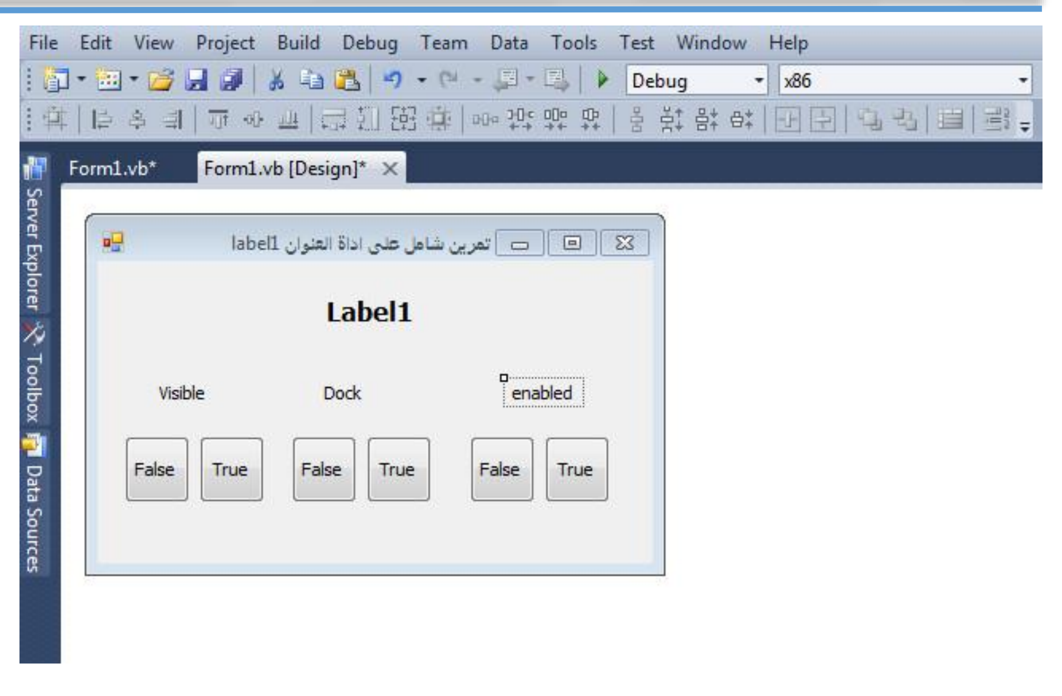
تمرين :

قم بانشاء برنامج يتحكم في الخواص التالية لاداة العنوان label1 وذلك من خلال انشاء ازرار مسماه تتحكم بكل خاصية :

- Locked
- Dock
- Visible

اسم الاداة	الخاصية	القيمة
Label1	Name	Lbl1
	Text	عنوان الاداة
	Font	Tahoma; 12pt; style=Bold
Label2	Name	Lbl2
	Text	Enabled
Label3	Name	Lbl3
	Text	Dock
Label4	Name	Lbl4
	Text	Visible
Button1	Name	Btn1
	Text	True
Button2	Name	Btn2
	Text	False
Button3	Name	Btn3
	Text	Right
Button4	Name	Btn4
	Text	Lift
Button5	Name	Btn5
	Text	True
Button6	Name	Btn6
	Text	False

انظر (الشكل٢٢-٣) ولاحظ شكل النموذج بعد اضافة الاداوات



(الشكل٢٢-٣) يوضح انشاء برنامج للتحكم بخصائص اداة العنوان **label1**

- ثم قم بادراج الكود المتعلق بالبرنامج في نافذة محرر الاكواد

Code	**Visual Basic.Net**

```
Public Class Form1

    Private Sub btn1_Click (By Val sender As System .Object, By Val e As System. EventArgs)
Handles btn1.Click
        lbl1.Enabled = True

'خاصية التمكين مفعلة
    End Sub

    Private Sub Button1_Click (By Val sender As System Object, By Val e As System. EventArgs)
Handles Button1.Click
        lbl1.Enabled = False

'خاصية التمكين غيرمفعلة
    End Sub

    Private Sub Button3_Click (By Val sender As System .Object, By Val e As System. EventArgs)
Handles Button3.Click
        lbl1.Dock = Dock Style. Right

'الملى لليمين
    End Sub

    Private Sub Button2_Click (By Val sender As System. Object, By Val e As System. EventArgs)
Handles Button2.Click
        lbl1.Dock = Dock Style. Left
'الملى لليسار
    End Sub

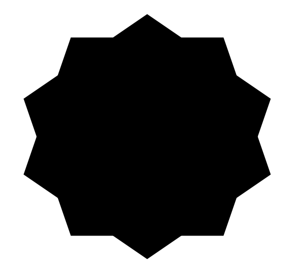
    Private Sub Button5_Click (By Val sender As System. Object, By Val e As System. EventArgs)
Handles Button5.Click
        lbl1.Visible = True
'اداة العنوان ظاهرة
```

```
    End Sub

    Private Sub Button4_Click (By Val sender As System. Object, By Val e As System. EventArgs)
Handles Button4.Click
        lbl1.Visible = False
'اداة العنوان مخفية
    End Sub

    Private Sub Form1_Load (By Val sender As System. Object, By Val e As System. EventArgs)
Handles My Base. Load

    End Sub
End Class
```



- اداة مربع النص

abl TextBox

استخدامها	تستخدم هذه الاداة لادخال البيانات ومن ثم استخدام تلك البيانات استخدام محدد مثلا أن نحتاج اداة Textbox لادخال درجات الطلاب سيكون نوع البيانات المدخل هنا عبارة عن ارقام وان كان الغرض منها ادخال البيانات المتعلقة باسماء الطلاب ستكون نوع البيانات المدخلة هي نصوص ...الخ .
مثال	قم بانشاء برنامج يقوم بادخال بيانات تسجيل درجات الطلاب في مربع نص Textbox واضف اليها مربع نص لتوصيفها باسم "درجات الطالب" كما (بالشكل السابق ١٢-٣):

خصائص اداة مربع النص Textbox

تمتلك اداة مربع النص الخواص التالي:

- Backcolor,BorderStyle,Cursor,Font,Forecolor,Right To lift,Text,TextAlign,Enabled,Locked,Dock,Size, visible, Location

وجميع هذه الخواص تم شرحها في اداة العنوان label1 ولكن هناك بعض الخواص الاخرى التي لم نتحدث عنها سابقاً سنتحدث عنها الان في اداة textbox سنقوم بشرحها الان بشكل مفصل :

الخاصية	القيمة	الشرح
- Character casing		• تتيح لك هذه الخاصية التحكم بحالة الاحرف التي تقوم بادخالها في مربع النص Text Box اما كبيرة اوصغيرة أو اعتيادية ولاتنطبق هذه الخاصية مع الاحرف العربية حيث انها تنطبق فقط على الاحرف الانجليزية انظر (الشكل ٢٠-٣)

الخاصية	القيمة	الوصف
	– Normal – Upper – Lower	(الشكل٢٠–٣) يوضح الخاصية **character casing** – ابقاء الحروف على حالتها . – تحويل الحروف الى كبيرة. – تحويل الحروف الى صغيرة.
– **Hide selection**	– True – False	• تمكنك هذه الخاصية من التحكم في اظهار واخفاء التظليل على الكلمات او النصوص او البيانات الاخرى المتواجدة في مربع النص **textbox1** بعد الانتقال الى ادوات اخرى . – ابقاء النص مظللاً بعد الانتقال من مربع النص الى مكان اخر في النموذج انظر (الشكل٢٠–٣): (الشكل٢٠–٣) يوضح ابقاء تظليل النص بعد الانتقال – اخفاء تظليل النص بعد الانتقال من مربع النص الى مكان اخر في النموذج .
– **Max Length**	– ٠	• تمكنك هذه الخاصية من التحكم في الحد الاقصى للاحرف المدخلة في مربع النص **Textbox1** اثناء التصميم والتنفيذ معاً . – بهذه الحالة لايمكن التقيد بحد معين لحالة عدد الاحرف في مربع النص **Textbox1** .

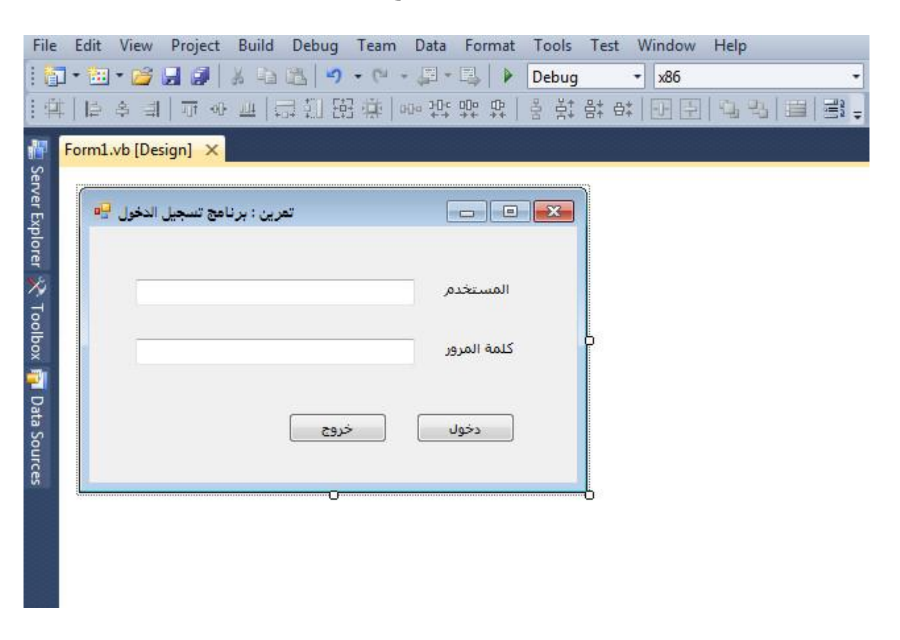
Multiline –	– عدد صحيح – True – False	– وبهذه الحالة تستطيع وضح حد معين لعدد الاحرف في مربع النص Textbox1 . • وتسمى هذه الميزة بميزة تعدد الاسطر أي بمعنى تمكنك من عرض عدد الاسطر في مربع النص Textbox1 بحيث يمكنك من تحديد عدد الاسطر الظاهرة في مربع النص . – أن مربع النص متعدد الاسطر . – أن مربع النص غير متعدد .
Password Char –	– * – !	• تمكنك هذه الخاصية من ترميز الكتابة الظاهرة في مربع النص بعدة اشكال أو رموز سرية لايستطيع أن يفهمها المستخدم العادي وتستخدم غالباً هذه الخاصية في نماذج وشاشات الدخول التي تمكنك من الدخول الى الانظمة . – اظهار رمز النجمة بدلاً من اظهار الحرف نفسه . انظر (الشكل ٢١-٣) : خصائص اداة مربع النص Textbox المستخدم user كلمة المرور **** (الشكل ٢١-٣) يوضح اظهار رمز* بدلا من الاحرف في مربع كلمة المرور . – اظهار علامة التعجب بدلا من الاحرف.
Read Only –	– True – False	• تمكنك هذه الخاصية من جعل النصوص المدخلة في مربع النص Textbox غير قابلة للتعديل وانما للقراءة فقط . – أن المستخدم لن يتمكن من تعديل النصوص في مربع النص . – أن المستخدم متاح له امكانية التعديل في النصوص المدخلة بمربع النص .

تمرين:

قم بانشاء برنامج تسجيل دخول يتكون من اداة العنوان **label** و اداة مربع النص **Text box** وزرين **Button** احداهما دخول والاخر ينفذ الخروج وقم بتغيير بعض الخواص الموجودة في الادوات المضافة واجعل قيمة **password Char** = * واجعل البرنامج يقوم بالتحقق من البيانات المدخلة في مربعات النص .

الخاصية	القيمة	الشرح
Label1	Name	Lbl1
	Text	"المستخدم"
Label2	Name	Lbl2
	Text	"كلمة المرور"
Textbox1	Name	Txt1
Textbox2	Name	Txt2
Button1	Name	Btn1
	Text	"دخول"
Button2	Name	Btn2
	Text	"خروج"

انظر (الشكل٢٢-٣) ولاحظ فيه النموذج بعد اضافة الادوات اليه :



(الشكل ٢٢-٣) يوضح شكل برنامج تسجيل الدخول بعد اضافة الادوات

بعد أن قمنا ببناء شكل النموذج نقوم الان باضافة الاكواد في نافذة تحرير الشفرة :

- ادخال الكود المتعلق بالتحقق من النصوص المدخلة في مربع نص المستخدم **txt2 & txt1** في **Btn1** نفرض اننا سنستخدم هنا الجملة الشرطية **IF…then** للتحقق من مدى البيانات المدخلة في مربع النص **txt2 & txt1** بمعنى اذا كانت البيانات النصية المدخلة بـ **txt1** هي **"user"** والبيانات المدخلة بـ **txt2** هي **"pass123"** تظهر لنا رسالة نصية تخبرنا بان البيانات المدخلة صحيحة وفي حالة لم يتحقق ذلك تظهر لنا رسالة نصية تخبرنا بان البيانات المدخلة خاطئة :

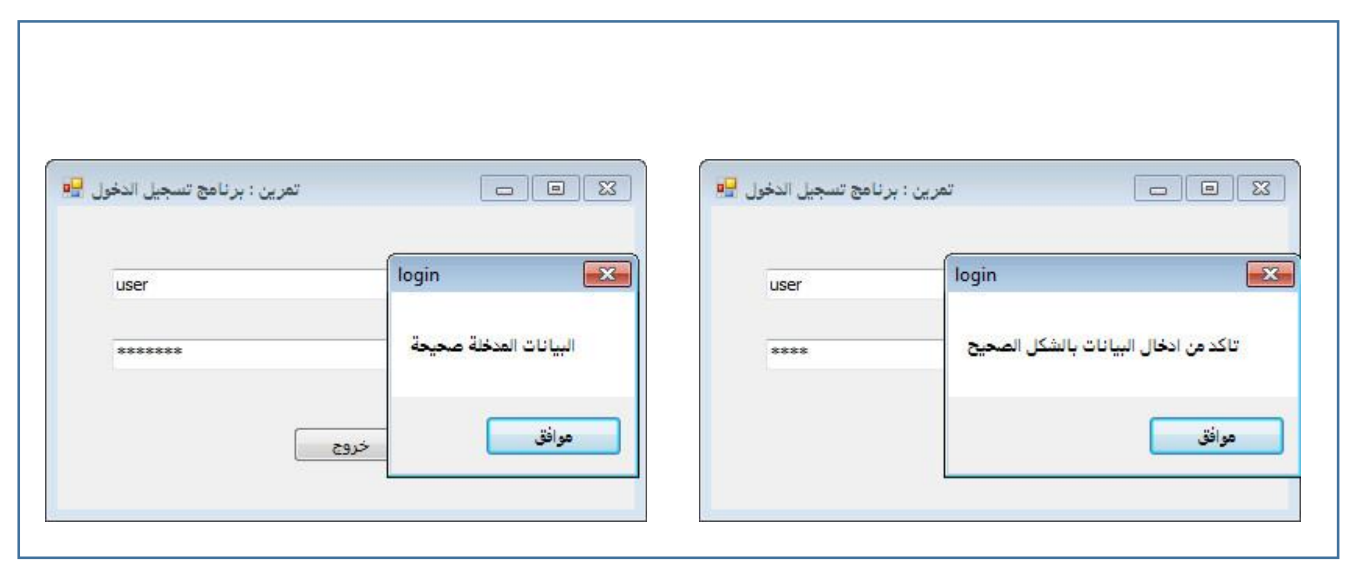
Visual Basic.Net — **Code**

```
Public Class Form1

    Private Sub btn1_Click (By Val sender As System .Object, By Val e As System. EventArgs)
Handles btn1.Click

If txt1.Text = "user" And txt2.Text = "pass123" Then
            MsgBox("البيانات المدخلة صحيحة")
        Else
            MsgBox("تاكد من ادخال البيانات بالشكل الصحيح")
        End If
    End Sub
```

انظر (الشكل٢٣-٣) يوضح تنفيذ الكود السابق :



(الشكل٢٣-٣) يوضح تنفيذ الكود السابق

- بعد أن قمنا بادخال كود التحقق من البيانات المدخلة في مربعات النصوص txt1 & txt2 نقوم الان بادخال كود الخروج في الزر btn2

Visual Basic.Net — **Code**

```
Public Class Form1

    Private Sub btn2_Click (By Val sender As System .Object, By Val e As System. EventArgs)
Handles btn2.Click

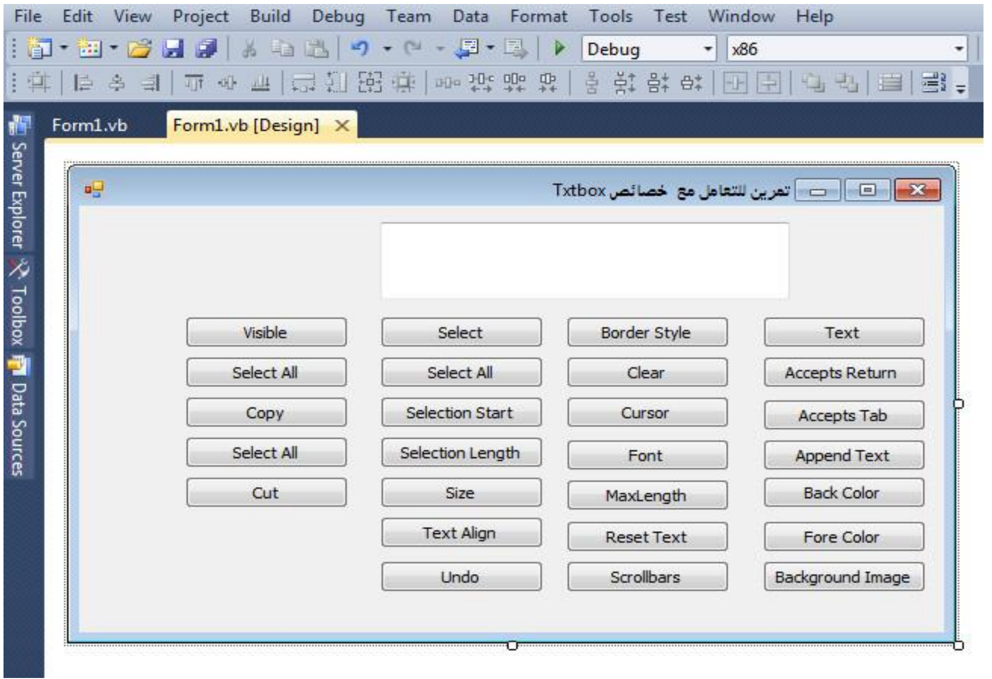
end
    End Sub
```

تمرين :

قم بانشاء برنامج له القدرة بالتعامل مع خصائص textbox التالية :

- **Text, Accepts Return, Accepts Tab, Append Text, Back Color, Fore Color, Background Image, Border Style, Clear, Cursor, Font, MaxLength, Reset Text, Scrollbars, Select, Select All, Selection Start, Selection Length, Size, Text Align, Undo, Visible, Select All, Copy, Select All, Cut**

- اولاً : نقوم ببناء شكل البرنامج ونقوم بانشاء ازرار بعدد الخصائص التي ذكرناها وتسميتها باسم تلك الخصائص انظر (الشكل ٢٤-٣) ولاحظ النموذج بعد اضافة الادوات اليه :



(الشكل ٢٤-٣) يوضح شكل برنامج التعامل مع خصائص textbox

• ثانياً : بعد أن قمنا ببناء شكل البرنامج نقوم الان بتركيب الاكواد المتعلقة بالخصائص :

Visual Basic.Net	Code

```
Public Class Form1

    Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button5.Click
        TextBox1.Clear()
        'مسح النصوص في مربع النص
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
        TextBox1.Text = "بكم يرحب الشميري المهندس"
        'عرض رسالة ترحيبية
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button2.Click
        TextBox1.AcceptsReturn = True
        ' سوف تمكن من الرجوع وانشاء سطر جديد

    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button4.Click
        TextBox1.AcceptsTab = True
        'سوف يمكن من قبول الضغط على الزر التاب والانتقال
    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button3.Click
        TextBox1.AppendText(vbCrLf & "اخرى مرة بكم نرحب")
        'سوف يتم دمج عبارة نرحب بكم مرة اخرى مع العبارة الموجودة  بالسطر السابق
    End Sub

    Private Sub Button24_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button24.Click
        TextBox1.BackColor = Color.Blue
        'تمكن من تغيير لون خلفية النص
    End Sub

    Private Sub Button8_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button8.Click
        TextBox1.ForeColor = Color.Black
        'سوف تمكن من تغيير لون الخط
    End Sub

    Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button7.Click
        TextBox1.BackgroundImage = Image.FromFile(Application.StartupPath & "\vb.Net.jpg")
        'سوف  تمكن من تغيير خلفية النص
    End Sub

    Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button6.Click
```

```
        TextBox1.BorderStyle = BorderStyle.FixedSingle
        'الحدود نمط تغيير
    End Sub

    Private Sub Button23_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button23.Click
        TextBox1.Cursor = Cursors.Hand
        'الفارة مؤشر تغيير
    End Sub

    Private Sub Button16_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button16.Click
        TextBox1.Font = New Drawing.Font("Comic Sans MS", 12, FontStyle.Bold, GraphicsUnit.Pixel)
        ' الخط خصائص تغيير
    End Sub

    Private Sub Button15_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button15.Click
        TextBox1.MaxLength = 3000
        'الحروف لعدد الاقصى الحد تحديد
    End Sub

    Private Sub Button14_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button14.Click
        TextBox1.ResetText()
        'الافتراضية القيمة الى النص ارجاع
    End Sub

    Private Sub Button13_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button13.Click
        TextBox1.ScrollBars = ScrollBars.Vertical
        'انزلاق شريط اضافة
    End Sub

    Private Sub Button22_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button22.Click
        TextBox1.Select(0, 11)
        'حرف12بطول البداية من النص تظليل
    End Sub

    Private Sub Button12_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button12.Click
        TextBox1.SelectAll()
        'بالكامل النص تظليل
    End Sub

    Private Sub Button11_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button11.Click
        TextBox1.SelectionStart = 0
        'حرف اول من التظليل بدء
    End Sub

    Private Sub Button10_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button10.Click
        TextBox1.SelectionLength = 11
        'المظلل النص طول
    End Sub

    Private Sub Button9_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button9.Click
        TextBox1.Size = New Size(300, 400)
```

```
        ' النص مربع حجم تغيير
    End Sub

    Private Sub Button21_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button21.Click
        TextBox1.TextAlign = HorizontalAlignment.Center
        'النص مربع منتصف في النص عرض
    End Sub

    Private Sub Button20_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button20.Click
        TextBox1.Undo()
        'النص لمربع تحرير اخر عن التراجع
    End Sub

    Private Sub Button19_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button19.Click
        TextBox1.Visible = True
        'النص مربع بظهور التحكم
    End Sub

    Private Sub Button18_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button18.Click
        TextBox1.SelectAll()
        'النص مربع مايحتويه كل تظليل
    End Sub

    Private Sub Button17_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button17.Click
        TextBox1.Copy()
        'النص مربع داخل المظلل الجزء نسخ
    End Sub

    Private Sub Button25_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button25.Click
        TextBox1.SelectAll()
        'النص مربع مايحتويه كل تظليل
    End Sub

    Private Sub Button26_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button26.Click
        TextBox1.Cut()
        'النص مربع داخل المظلل الجزء قص
    End Sub
End Class
```

- ## اداة زر

Button

عزيزي القارئ ستلاحظ أن الادوات تشابهة في معظم خصائص **Button** كما تعلم عزيزي القارئ أن الزر **Button** يستخدم لتنفيذ امر محدد يرغب المبرمج في تنفيذه ويمكن لك أن تتعامل مع كافة خصائص **Button** سنذكرك عزيزي القارئ بتلك الخصائص وانه بامكانك التعامل مثل ما تعاملنا مع الادوات السابقة :

Fore ، Font ، Enabled ، Dock ، Curser ، Context menu ، Cause Validate ، Back color ، Color، Locked ، Size ،Tab Index ، Tab Stop ، Text، Text Align .

فلن نقوم حاليا بشرح هذه الخصائص حتى لانضيع الوقت اكثر في تكرارها .

- اداة مربع الاختيار CheckBox

استخدامها	تستخدم هذه الاداة لاختيار امر من مجموعة من الاوامر أو للقيام بعملية تصويب خيار محدد ستلاحظ عزيزي القارئ أنه يظهر في مربع الاختيار علامة صح (**True**) ويمكن أن تختفي عن النقر على المربع مرة اخرى وذلك يعني الامر خطأ (**False**)
مثال	قم بانشاء برنامج يقوم باختيار نوع اللون الظاهر في خلفية اداة العنوان هل هو ازرق أم اخضر اذا تم اختيار اللون بالشكل الصحيح تظهر لك رسالة نصية تخبرك بذلك والعكس صحيح .

عزيزي القارئ هناك بعض الخصائص التي يتكرر ذكرها في الادوات السابقة ك الاتي :

Fore ، Font ، Enabled ، Dock ، Curser ، Context menu ، Cause Validate ، Back color ، Color، Locked ، Size ،Tab Index ، Tab Stop ، Text، Text Align .

فلن نقوم حاليا بشرح هذه الخصائص حتى لانضيع الوقت اكثر في تكرارها ، ولكننا سنكتفي بشرح خصائص اخرى لم نذكرها من ضمن تلك الخصائص وهي كالاتي :

الخاصية	القيمة	الشرح
– **Appearance**	– Normal – Button	• تمكنك هذه الخاصية من التحكم بشكل ظهور مربع الاختيار فتوجد قيميتين تمثل كل شكل على حده لمربع الاختيار **check box** . – معناه أن مربع الاختيار سيظهر بالشكل العادي له . – وهنا ستلاحظ ان مربع الاختيار سياخذ شكل زر تنفيذ الاوامر انظر (الشكل ٢٤-٣) .

اعداد / محمد حازم الشميري

		(الشكل ٢٤-٣)يوضح شكل مربع الاختيار
- Auto Check	- True - False	خاصيةالاختيار التلقائي هي عبارة عن خاصية تمكنك من تصويب مربع الاختيار ووضع علامة صح (True) بشكل تلقائي. - تمكنك هذه الخاصية من وضع علامة الاختيار (true) عند النقر بشكل تلقائي على مربع الاختيار. - هنا لن تمكنك من الاختيار مهما قمت بعملية النقر والاختيار في المربع .
- Check Align		• تمكنك هذه الخاصية من تنسيق شكل الاختيار وهذه الخاصية تشبه نوعا ما الخاصية Dock .
- Checked	- True - False	• تمكنك هذه الخاصية للقيام بعملية التاكد من قيام المستخدم بالاختيار في مربع الاختيار ومعرفة هل هو (True) أم (False) . - المستخدم قام بالاختيار . - المستخدم لم يقم بالاختيار.
- Check State	- Unchecked - Checked - Indeterminate	• تمكنك هذه الخاصية من معرفة حالة الاختيار . - لم يقم المستخدم بالاختيار . - قام المستخدم بالاختيار. - غير محدد .
- Three State	- True - False	• تمكنك هذه الخاصية من القيام بعملية اضافة حالة ثالثة للاختيار اي بمعنى انه الحالة الثالثة مثلا لن نقوم باختيارها . - معناه انه يوجد ثلاث حالات لمربع الاختيار. - اما هنا فتوجد حالتان فقط اما صواب أو بدونها .

ملاحظة :

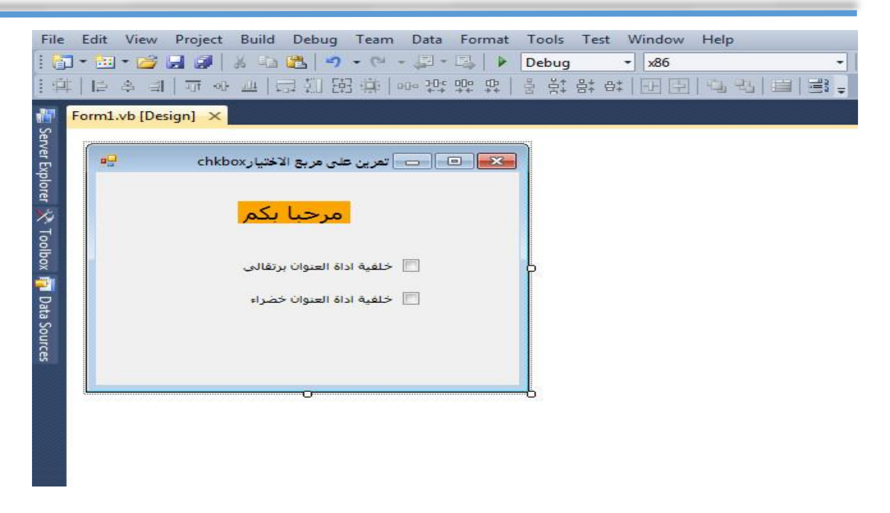
حتى تتمكن من استخدام الخاصية check state يجب عليك اولا أن تقوم بتفعيل الخاصية Three state وتعطيها قيمة True .

تمرين :

قم بانشاء برنامج يقوم باختيار نوع اللون الظاهر في خلفية اداة العنوان هل هو ازرق أم اخضر اذا تم اختيار اللون بالشكل الصحيح تظهر لك رسالة نصية تخبرك بذلك والعكس صحيح .

الخاصية	القيمة	الشرح
Label1	Name Text Font Back color	Lbl1 "مرحبا بكم" "Tahoma; 14.25pt" "Orange"
CheckBox1	Name Text	Chkbox1 " خلفية اداة العنوان برتقالي"
CheckBox2	Name Text	Chkbox2 " خلفية اداة العنوان خضراء"

انظر (الشكل ٢٥-٣) ولاحظ النموذج بعد اضافة الادوات اليه :



(الشكل ٢٥-٣) بعد اضافة الادوات اليه

– بعد أن قمنا ببناء شكل النموذج نقوم الان بتركيب الاكواد المتعلقة بالبرنامج وسنستخدم حاليا الجملة **if...then** للتحقق من الاختيار في مربع الاختيار ندخل الكود المتعلق بمربع الاختيار الاول :

Visual Basic.Net — **Code**

```
Public Class Form1
Private Sub chkbox1_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles chkbox1.CheckedChanged
        If chkbox1.Checked = True Then
            MsgBox("صحيح اختيارك")
            'صحيح اختيارك صائب الاختيار وان طالما
        End If
    End Sub
```

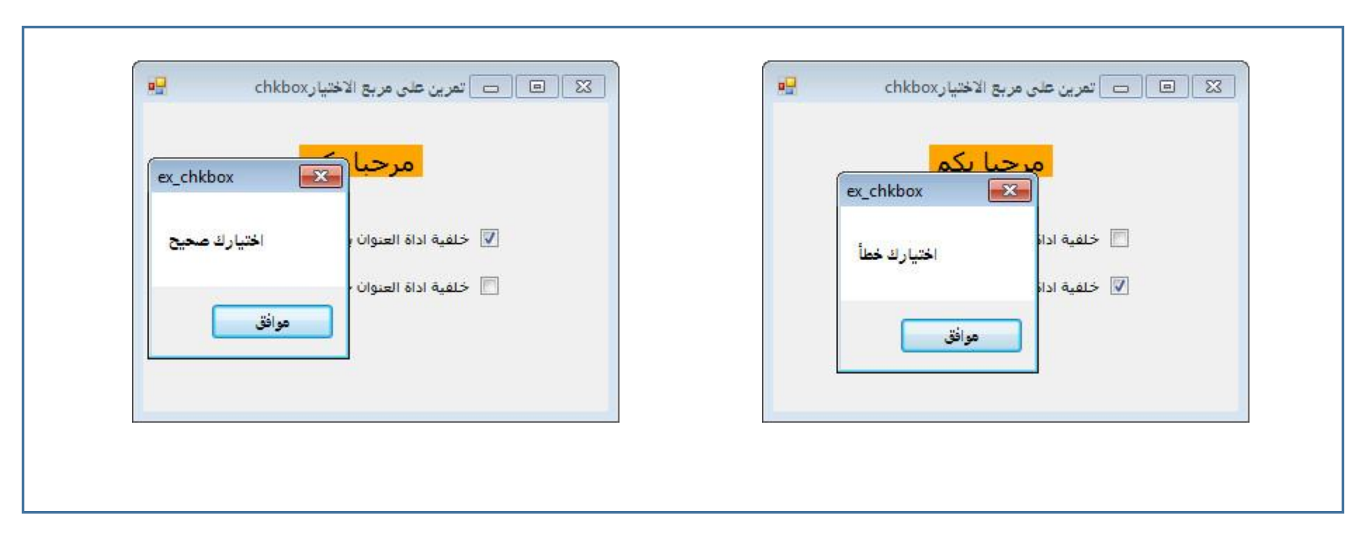
– ثم نقوم بادخال الكود المتعلق بمربع الاختيار الثاني :

Visual Basic.Net — **Code**

```
Public Class Form1

Private Sub chkbox2_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles chkbox2.CheckedChanged
        If chkbox2.Checked = True Then
            MsgBox("خطأ اختيارك")
            'صحيح اختيارك صائب الاختيار وان طالما
        End If
    End Sub
```

انظر (الشكل ٢٦-٣) ولاحظ النموذج بعد التنفيذ :



(الشكل٢٦-٣) يوضح تنفيذ البرنامج السابق

- اداة زر الاختيار **Radio button**

استخدامها	الاداة **Radio button** تعتبر اداة وحيدة الاختيار بمعنى انها تتيح للمستخدم اختيار خيار واحد من مجموعة خيارات حيث توجد بينها وبين اداة الاختيار السابقة **check box** حيث نجد ان في زر الاختيار **Radio button** لاتوجد الخاصية **Three State** التي تتيح للمستخدم امكانية تعدد حالات الاختيار .
مثال	قم بانشاء برنامج يقوم بتغيير خاصية الالوان لخلفية اداة العنوان .

عزيزي القارئ هناك بعض الخصائص التي يتكرر ذكرها في الادوات السابقة كـ الاتي :

Back color ، Cause Validate ، Context menu ، Curser ، Dock ، Enabled ، Font ، Fore Color، Locked ، Size ، Tab Index، Tab Stop ، Text، Text Align .

ملاحظة :

عزيزي القارئ يوجد فرق بين اداة مربع الاختيار Check box ومابين زر الاختيار Radio button حيث أن الخاصية المتعلقة بـ three state والمتوفرة في مربع الاختيار check box غير متوفرة في اداة زر الاختيار Radio button كما انه يمكنك من خلال مربع الاختيار القدرة على اختيار اكثر من حالة اختيار بينما في اداة زر الاختيار Radio button لايمكنك سوى اختيار حالة واحدة فقط وسيتضح لك من خلال (الشكل ٢٧-٣) :



(الشكل ٢٧–٣) يوضح الفرق بين مربع الاختيار وزر الاختيار

تمرين :

قم بانشاء برنامج يقوم بالتحكم بخاصية تغيير لون الخط لاداة العنوان **label** واضف الى البرنامج الادوات التالية :

الخاصية	القيمة	الشرح
Label1	Name Text	Lbl1 "اللون"
GroupBox1	Name Text	Group1 "لون الخط"
Radio button1	Name Text	Rd1 "ازرق"
Radio button2	Name Text	Rd2 "اخضر"
Radio button3	Name Text	Rd3 "احمر"
Textbox1	Name	Txt1
Button1	Name Text	Btn1 "عرض النص"

- اولاً : سنقوم ببناء شكل البرنامج وباحتوى ازرار الاختيار في اداة **GroupBox1** انظر (الشكل ٢٨-٣) :



(الشكل ٢٨-٣) يوضح شكل برنامج الالوان

- ثانياً : بعد أن قمنا ببناء شكل البرنامج نقوم بالاتي :
  - نقوم بتركيب الكود المتعلق بعرض النص المدخل في مربع النص أن يتم عرضه في اداة العنوان :

Visual Basic.Net **Code**

```
Public Class Form1

Private Sub btn1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btn1.Click
        lbl1.Text = txt1.Text

    End Sub
```

- نقوم بادخال الكود المتعلق بتغيير لون اداة العنوان عند اختيار الزر الذي يمثل اللون الازرق في **rd1** وذلك بالنقر المزدوج على اداة زر الاختيار **Rd1** لكي نقوم بتركيب الكود في الحدث :

Visual Basic.Net **Code**

```
Public Class Form1

Private Sub rd1_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
rd1.CheckedChanged
        lbl1.ForeColor= Color.Blue


    End Sub
```

- نقوم بادخال الكود المتعلق بتغيير لون اداة العنوان عند اختيار الزر الذي يمثل اللون الاخضر في **rd2** وذلك بالنقر المزدوج على اداة زر الاختيار **Rd2** لكي نقوم بتركيب الكود في الحدث :

Visual Basic.Net **Code**

```
Public Class Form1

Private Sub rd2_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
rd2.CheckedChanged
        lbl1.ForeColor= Color.Green


    End Sub
```

– نقوم بادخال الكود المتعلق بتغيير لون اداة العنوان عند اختيار الزر الذي يمثل اللون الاحمر في rd3 وذلك بالنقر المزدوج على اداة زر الاختيار Rd3 لكي نقوم بتركيب الكود في الحدث :

Visual Basic.Net **Code**

```
Public Class Form1

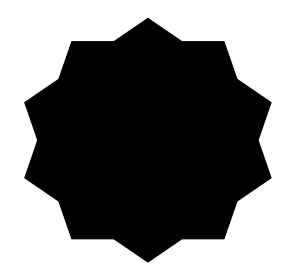
Private Sub rd3_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
rd3.CheckedChanged
        lbl1. ForeColor= Color.Red


    End Sub
End Sub
```

– بعد أن قمنا بتركيب الاكواد المتعلقة بالبرنامج نقوم الان بتنفيذ البرنامج وتشغيله ومن ثم نقوم بادخال نص ترحيبي في مربع النص ونقوم بالضغط على الزر عرض النص ليقوم بنقل النص او عرضه في اداة العنوان ومن ثم نقوم باختيار ازرار الاختيار والتنقل فيما بينها ستلاحظ انه يقوم بتغيير لون الخط في اداة العنوان لاحظ (الشكل ٢٩–٣) :



(الشكل ٢٩–٣) يوضح شكل البرنامج بعد القيام باختيار زر الاختيار "ازرق"



• اداة القائمة list box



تعتبر اداة القائمة من الادوات الهامه كثيرة الاستخدام في الحياة العملية والتي يحتاجها بكثرة المبرمج في تصميم برامجه وتطويرها

استخدامها	تستخدم اداة القائمة **list box** لعرض العناصر فيها بشكل مسرد ومرتب بحيث تمكن المستخدم من اختيار تلك العناصر والتعامل معها اما بالاضافة والتعديل او الحذف او جعل احدى العناصر المحددة أو التي تم اختيارها من تنفيذ امر معين كما انه بامكان المستخدم القدرة على ختيار عنصر واحد او اكثر بالقائمة .
مثال	قم بانشاء برنامج يقوم بعرض أو اضافة اسماء الادارات أو الاقسام الموجودة في الشركة .

عزيزي القارئ هناك بعض الخصائص التي يتكرر ذكرها في الادوات السابقة كـ الاتي :

Back color ، Cause Validate ، Context menu ، Curser ، Dock ، Enabled ، Font ، Fore ، Color، Locked ، Size ، Tab Index، Tab Stop .

وهناك من الخصائص التي لم نذكرها سابقاً سنشرحها الان انظر الجدول الموجود اسفل :

الخاصية	القيمة	الشرح
– **Multicolumn**		• تمكنك هذه الخاصية من التحكم من عرض العمود في القائمة **list box** حتى تتمكن من الاستفادة من تطبيق هذه الخاصية يجب أن تعطى لها القيمة **True** انظر (الشكل٣٠-٣) ItemHeight 14 MultiColumn True ScrollAlwaysVis True SelectionMode False Sorted False TabIndex 0 TabStop True UseTabStops True (الشكل٣٠-٣) يوضح اعطاء القيمة **True** للخاصية

- ColumnWidht	- صفر - عدد صحيح	- معناه أن النظام سوف يحدد عرض الاعمدة. - عرض العمود في القائمة متعدد الاعمدة .
- **Draw Mode**	- Normal - OwnerDrawFixed - OwnerDrawVariable	• اما عن خاصية شكل الظهور التي من التحكم بشكل ظهور القائمة **list box** وتتيح لك المجال اما أن تقوم بتركها لتحكم بها من قبل النظام أم انك تريد أن تتحكم بها انت . - معناه أن النظام سوف يحدد شكل القائمة . - معناه انت من سيقوم بتغيير خاصية الشكل عرض العناصر (ارتفاعها) مع ملاحظة ان الارتفاع سيكون هو نفسه لجميع العناصر وذلك من خلال الخاصية **height** - أن المستخدم من يحدد شكل العناصر كما انه بامكانك اعطاء ارتفاعات مختلفة للعناصر .
- **Item Height**		• خاصية ارتفاع العناصر تمكنك هذه الخاصية من تحديد ارتفاع العنصر في القائمة .

ملاحظة :

- **عزيزي القارئ ستلاحظ اننا لم نقم بشرح الخاصية data source & data bindings & display member في الادوات السابقة ولكننا سنتطرق لها في كتب لاحقه وبالذات في فصل مدخلك في برمجة قواعد البيانات .**
- **لكي تتمكن من استخدام هذه الخاصية** Item Height يجب عليك أن تقوم باختيار احدى القيم التالية من خاصية شكل الظهور OwnerDrawFixed أو OwnerDrawVariable
  - **ففي حال كانت خاصية الظهور هي** OwnerDrawFixed فان قيمة هذه الخاصية ستكون هي قيمة ارتفاع العنصر .
  - **اما في حال كانت قيمة خاصية الظهور هي** OwnerDrawVariable **ففي هذه الحالة ستكون هذه القيمة هي كـــ قيمة ارتفاع لاخر عنصر تمت اضافته .**

الخاصية	القيمة	الشرح
– Items		• خاصية العناصر : تمكنك هذه الخاصية من القدرة على اضافة العناصر التي ترغب باضافتها الى القائمة list box فيمكنك اضافة العناصر الى القائمة اما من خلال اضافتها يدويا من خلال الخاصية items الموجودة في نافذة الخصائص انظر (الشكل ٣١-٣) : (الشكل ٣١-٣)يوضح اختيار الخاصية items ومن ثم نقوم بادراج العناصر المراد اضافتها على سبيل المثال اذا رغبنا باضافة الادارات في الشركة انظر (الشكل ٣٢-٣) : (الشكل ٣٢-٣)يوضح اضافة العناصر يدوياً كما انك ستتعلم عزيزي القارئ كيف تقوم بالتعامل مع القائمة بشكل اكبر من المواضيع القادمة مثل : اضافة العناصر وتعديلها وحذفها من القائمة .
– Multicolumn	– True – False	• خاصية تعدد الاعمدة : تعتبر من الخواص الهامة في اداة القائمة list box حيث انها تمكنك من اضافة اكثر من عمود في القائمة . – القائمة متعددة الاعمدة . – القائمة غير متعددة الاعمدة (عمود واحد فقط) .

ملاحظة:

في حال أن قمت باختيار القيمة True لخاصية تعدد الاعمدة يعني ذلك عزيزي القارئ انك بحاجة للتحكم بالمسافة الفاصلة بين تلك الاعمدة أي عرض الاعمدة Column Width .

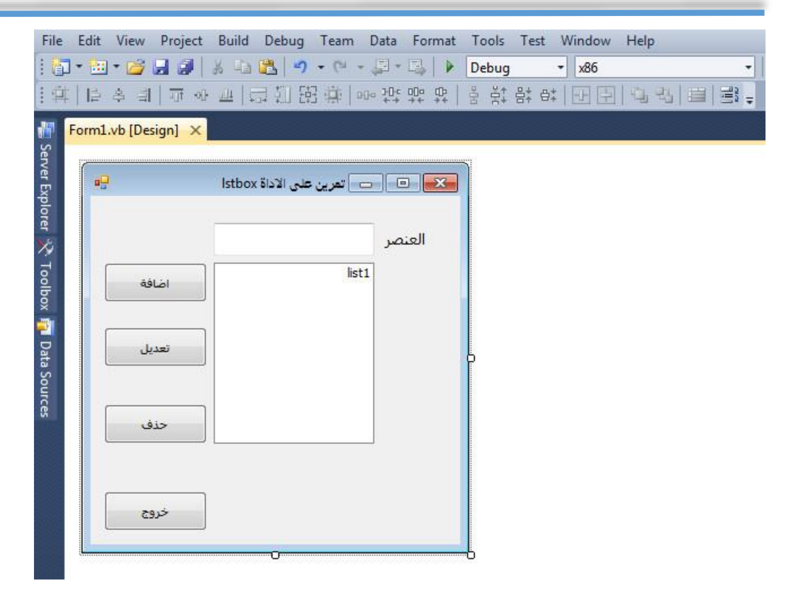
الخاصية	القيمة	الشرح
– ScrollAlwaysVisible		• الظهور الدائم لشريط التمرير : تمكنك هذه الخاصية من التحكم في ظهور شريط التمرير في الجانب الايمن للقائمة .
	– True	– امكانية عرض عدة اعمدة .
	– False	– الاكتفاء بعمود واحدفقط .
– Selection Mode		• خاصية طريقة الاختيار : تمكنك هذه الخاصية من تحديد الطريقة التي يمكنك من خلالها اختيار العناصر في القائمة .
	– None	– لايستطيع المستخدم تحديد أي عنصر .
	– One	– بامكان المستخدم تحديد عنصر واحد فقط .
	– MultiSimple	– تمكن المستخدم تحديداكثر من عنصر انظر (الشكل ٣٣-٣) : (الشكل٣٣-٣) يوضح خاصية **MultiSimple**
	– MultiExtended	– تمكن المستخدم تحديداكثر من عنصر كذلك .

– Sorted	– True – False	• خاصية الترتيب : تستخدم للقيام بالترتيب التلقائي للعناصر في القائمة . – ترتيب حسب الحروف الهجائية . – الترتيب حسب اولوية تحميل العناصر.

تمرين :

قم بانشاء برنامج يتحكم باضافة وتعديل وحذف العناصر من والى القائمة **list box** من خلال اضافة الادوات الاتية الموجود في الجدول بالاسفل :

الخاصية	القيمة	الشرح
Label1	Name Text	Lblitem "العنصر"
Textbox1	Name	Txtitem
List box1	Name	List1
Button1	Name Text	btnadd "اضافة"
Button2	Name Text	btnupdate "تعديل"
Button3	Name Text	btndelete "حذف"
Button4	Name Text	Btnexit "خروج"



(الشكل٣٤-٣) يوضح اضافة الادوات الى النموذج

- بعد أن قمنا ببناء شكل النموذج نقوم الان بتركيب الاكواد المتعلقة بالازرار :

اولاً : الكود المتعلق بزر اضافة العناصر المدخلة في مربع النص **Txtitem** الى القائمة **List1**

Visual Basic.Net	Code

```
Public Class Form1

Private Sub btnadd_Click(ByVal sender As System. Object, ByVal e As System.EventArgs) Handles
btnadd.Click
        If txtitem.Text <> "" Then
            list1.Items.Add(txtitem.Text)
            txtitem.ResetText()
            txtitem.Focus()
        End If

    End Sub

End Sub
```

ملاحظة:

- **يقوم المستخدم بادخال البيانات في مربع النص فمن خلال الكود السابق يقوم البرنامج بالتفحص في مربع النص بحال لم تدخل بيانات في مربع النص فانه لايضاف شي للقائمة اما في حالة تواجد بيانات داخل مربع النص سيقوم البرنامج باضافة النص أو البيانات الموجودة داخل مربع النص الى القائمة ثم يقوم البرنامج او الكود بتفريغ مربع النص من البيانات ثم يقوم بتفعيله فعلى سبيل المثال قم باضافة اسماء الادارات الاتية في مربع :**
  - **الادارة العامة**
  - **ادارة الانتاج**
  - **ادارة المبيعات**
  - **الادارة المالية**

في كل مرة تكتب اسم الادارة في مربع النص ويتم الضغط على الزر اضافة انظر (الشكل٣٥-٣)



(الشكل ٣٥-٣) يوضح اضافة العناصر الى القائمة

- ثانياً : ادخال الكود المتعلق بعرض عناصر القائمة في مربع النص من اجل التعديل فاننا سنتعامل بهذه الحالة مع الحدث SelectedIndexChanged في الاداة list1 من اجل تغيير العناصر المختارة والتي سيتم عرضها في مربع النص ولادخال الكود قم بالنقر المزدوج على اداة القائمة :

Visual Basic.Net — **Code**

```
Public Class Form1
Private Sub list1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles list1.SelectedIndexChanged
        txtitem.Text = list1.SelectedItem.ToString
    End Sub
End Sub
```

- ثالثاً : ادخال الكود المتعلق بزر التعديل والذي سيمكن المستخدم من القيام بعملية تعديل عناصر القائمة :

Visual Basic.Net — **Code**

```
Public Class Form1

Private Sub btnupdate_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnupdate.Click
        If list1.SelectedIndex <> -1 Then
            list1.Items.Remove(list1.SelectedIndex)
            list1.Items.Add(txtitem.Text)
            txtitem.ResetText()
        End If
    End Sub
End Sub
```

شرح الكود :

- كما تلاحظ عزيزي القارئ في العبارة الاولى سيقوم الكود من التحقق من امكانية تحديد عنصر ما في القائمة من خلال العبارة *SelectedIndex* ففي حال تبين انه لم يتم اختيار أي عنصر في القائمة باعطائه القيمة -١ للخاصية *SelectedIndex* *في حال عدم اختيار أي عنصر*.

- اما في العبارة الثانية Remove سيتم حذف العنصر المحدد من القائمة .

- اما في العبارة الثالثة سيتم اضافة البيانات المدخلة في مربع النص الى القائمة .

- اما في العبارة الاخيرة سيتم استعادة مربع النص لوضعه الطبيعي (افراغ) .

- رابعاً : ادخال الكود المتعلق بزر الحذف والذي من حلاله تستطيع حذف اي عنصر يتم تحديده في القائمة :

Visual Basic.Net **Code**

```
Public Class Form1

Private Sub btndelete_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btndelete.Click
        Dim s As String
        s = txtitem.Text & "هل انت متاكد من الحذف ؟ "
If MessageBox.Show(s ،" تحذير :فذح"،MessageBox.YesNo + MessageBox.IconQuestion) _
= DialogResult.Yes Then
            list1.Items.Remove(list1.SelectedIndex)
        End If
    End Sub
End Sub
```

شرح الكود :

- كما تلاحظ عزيزي القارئ في العبارة الاولى قمنا بتعريف المتغير X بنص رسالة الحذف و البيانات الظاهرة في مربع النص txtitem .

- اما في العبارة الثانية فقد استخدام الجملة الشرطية للتحقق من ناتج عرض دالة messagebox اذا كان ناتج مربع الحوار هو اختيار نعم هنا سيتم الانتقال بتنفيذ العنصر المحدد في القائمة .

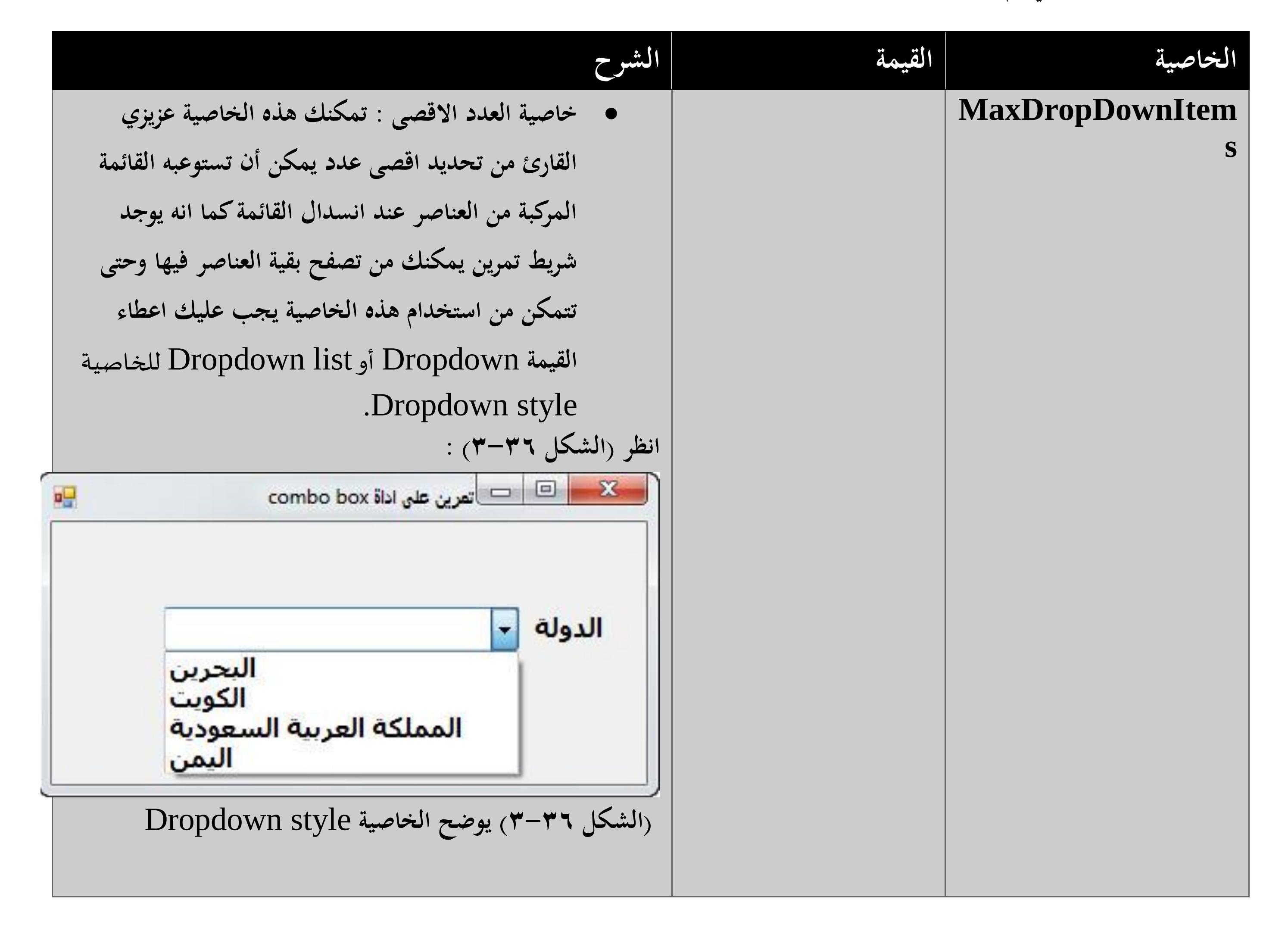
• اداة القائمة المركبة Comb box

ComboBox

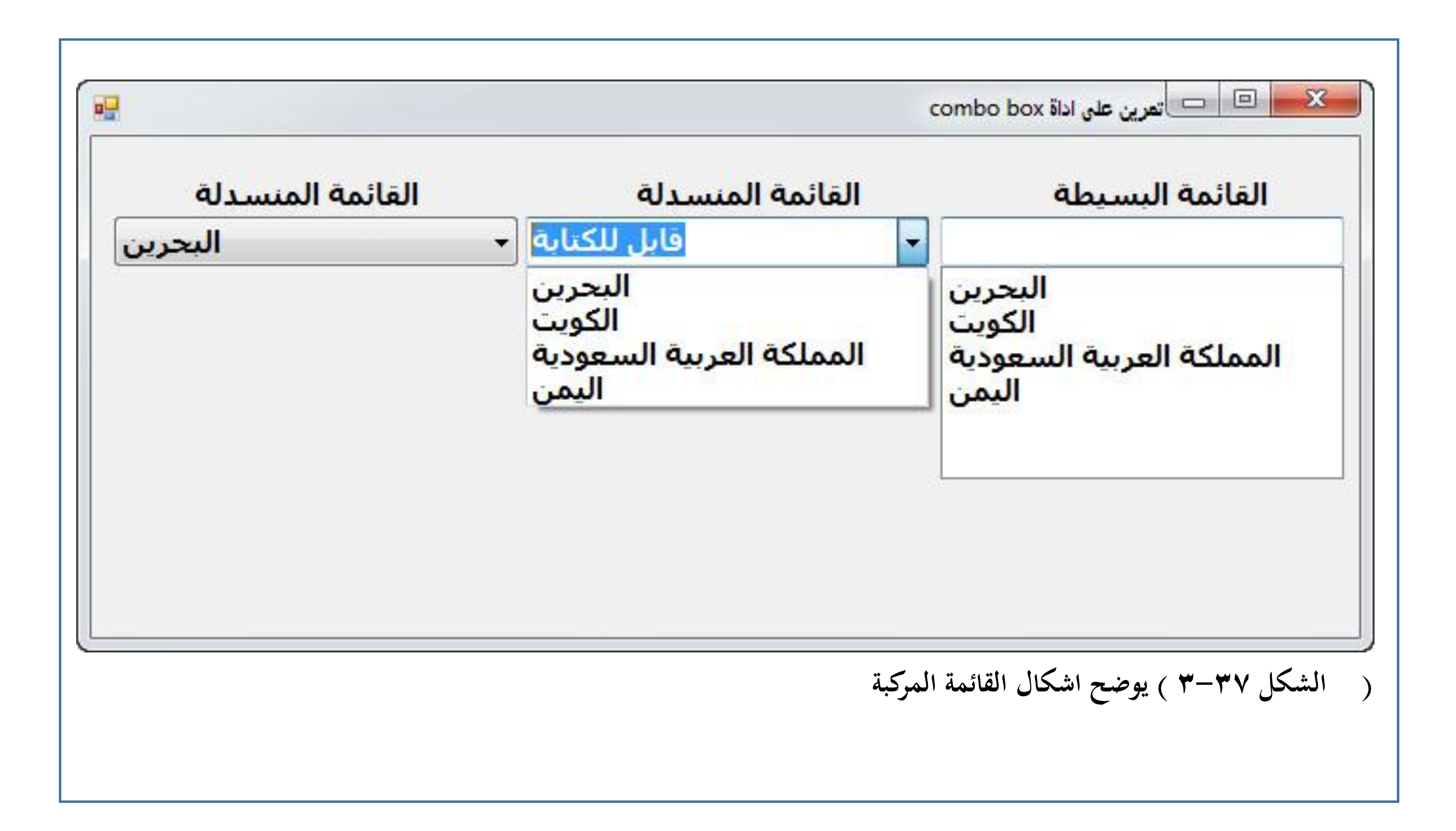
استخدامها	اداة القائمة المركبة **Combo box** استخدامها يتشابه لحد كبير مع اداة القائمة **list box** الا ان اداة القائمة المركبة لها بعض الاستخدامات الاخرى التي لا يمكن لاداة القائمة **list box** فان اداة القائمة المركبة يمكن لها أن تعرض عدد من العناصر وسردها بشكل افضل وتحتوي اكبر كم بشكل يوفر مساحة في النموذج .
مثال	أن تقوم بانشاء برنامج يقوم بعرض اسماء الدول العربية في القائمة المركبة .

عزيزي القارئ هناك بعض الخصائص المتعلقة بالقائمة المركبة قد تم ذكرها في اداة القائمة **list box** فلا داعي لذكرها مرة اخرى :

وهناك من الخصائص التي لم نذكرها سابقاً سنشرحها الان انظر الجدول الموجود اسفل :

الخاصية	القيمة	الشرح
MaxDropDownItems		• خاصية العدد الاقصى : تمكنك هذه الخاصية عزيزي القارئ من تحديد اقصى عدد يمكن أن تستوعبه القائمة المركبة من العناصر عند انسدال القائمة كما انه يوجد شريط تمرين يمكنك من تصفح بقية العناصر فيها وحتى تتمكن من استخدام هذه الخاصية يجب عليك اعطاء القيمة Dropdown أو Dropdown list للخاصية .Dropdown style انظر (الشكل ٣٦-٣) :  (الشكل ٣٦-٣) يوضح الخاصية Dropdown style

- style DropDown		• خاصية الشكل : تمكنك هذه الخاصية من اختيار الشكل المناسب لظهور القائمة المركبة .
	- simple	- يكون شكل الاداة المركبة فيها مشابهاً لمربع النص .
	- DropDown	- يكون شكل القائمة منسدلة مع امكانية التعديل والكتابة في مربع نص القائمة انظر الشكل السابق (٣٦-٣)
	- DropDownList	- يكون شكل هذه القائمة منسدلاً مع عدم امكانية الكتابة في مربع نص القائمة .



(الشكل ٣٧-٣) يوضح اشكال القائمة المركبة

تمرين :

قم بانشاء برنامج له القدرة بالتعامل مع اداة القائمة **combo box & list box** بحيث يقوم البرنامج بعرض اسماء الصفوف الدراسية في القائمة **combo box** وعند اختيار صف محدد من القائمة المركبة يتم عرض اسماء الطلاب في ذلك الصف وقم ببناء شكل البرنامج حسب الجدول الموجود بالاسفل :

الخاصية	القيمة	الشرح
Label1	Name	lblclasse
	Text	"الصف"
Combo box	Name	Cbo1
List box1	Name	List1

- قم باضافة بيانات الصفوف في اداة القائمة المركبة :

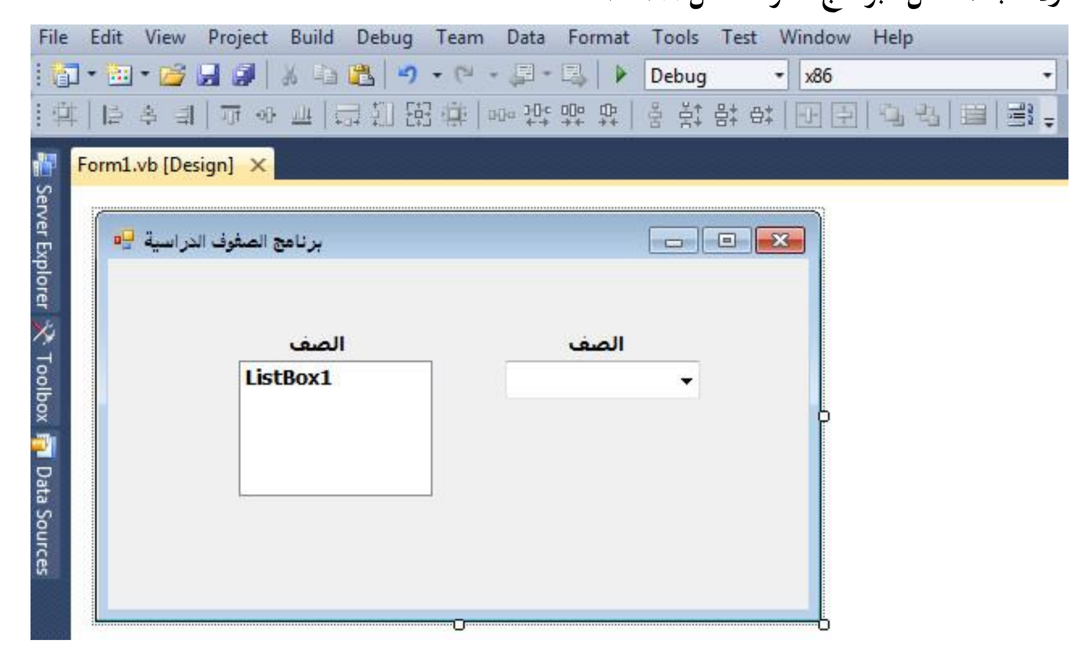
اول ابتدائي – ثاني ابتدائي .

- قم باضافة بيانات الطلاب في كل صف على حدة في اداة القائمة **list box** :

الصف	الطلاب
الاول ابتدائي	حمزة محمد الشميري
	حازم عبدالباسط الشميري
الثاني ابتدائي	عرفات محمد الشميري

بشرط أن تتم اضافة العناصر وعرضها من خلال الاكواد البرمجية :

- اولا : بناء شكل البرنامج انظر الشكل ٣٨-٣ :



الشكل ٣٨-٣ يوضح شكل البرنامج بعد اضافة الادوات

- ثانيا : القيام بتركيب كود اعلان اضافة بيانات الصفوف في حدث النموذج Form_load :

Visual Basic.Net **Code**

```
Public Class Form1

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load

        cbo1.Items.Add("ابتدائي اول")
        cbo1.Items.Add("ابتدائي ثاني")

    End Sub
End Class
```

- ثالثاً : نقوم بتركيب كود عرض اسماء الطلاب اثناء اختيار صف محدد من القائمة المركبة combo box يكتب الكود في الحدث selectedindexchanged المتعلق بالقائمة المركبة :

Visual Basic.Net **Code**

```
Public Class Form1

Private Sub cbo1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles cbo1.SelectedIndexChanged
        Select Case cbo1.SelectedIndex
            Case Is = 0

                list1.Items.Clear()

                list1.Items.Add("الشميري محمد حمزة")

                list1.Items.Add("الشميري عبدالباسط حازم")

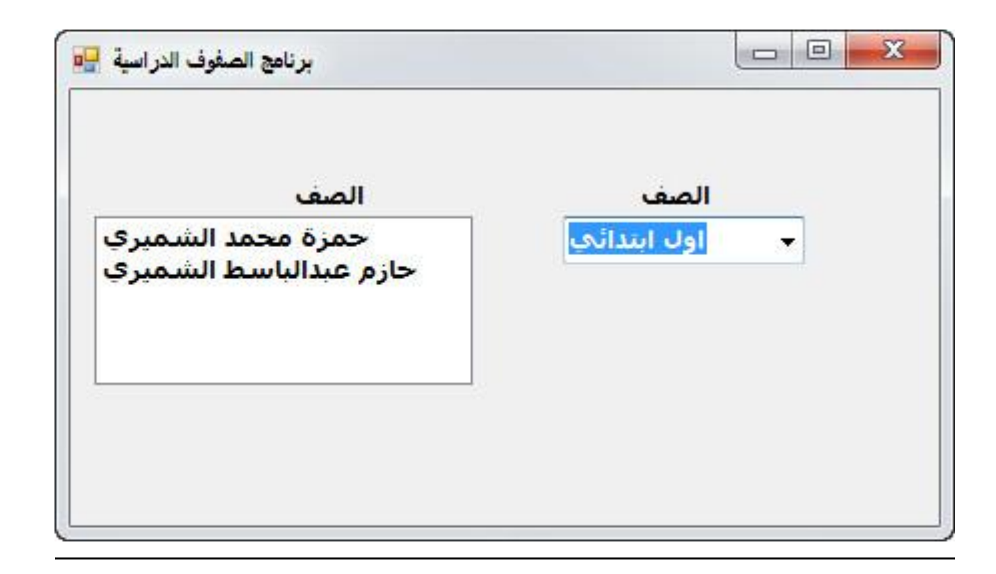
            Case Is = 1

                list1.Items.Clear()

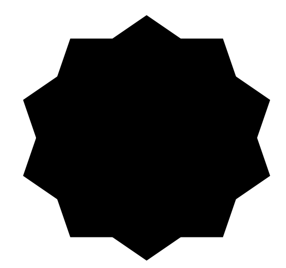
                list1.Items.Add("الشميري محمد عرفات")

        End Select


    End Sub
End Class
```



الشكل ٣٩-٣ يوضح اضافة العناصر والتحكم في عرضها



• اداة التوقيت Timer



تستخدم اداة التوقيت Timer في تنفيذ امر معين خلال فترة محددة كما انه يمكن من خلالها تزمين الاحداث والاوامر مثلا بحال انك ترغب من البرنامج أن يقوم بعرض رسالة ترحيبية بالصباح ورسالة ترحيبية بالمساء ماعليك الا أن تقوم بكتابة كود الرسالة الترحيبية في الحدث Tick() وتوجد لاداة التوقيت Timer خصائص تختلف عن الادوات السابقة التي ذكرناها ومن تلك الخصائص interval وهي تمثل الفترة بالملي ثانية (Millisecond) ويوجد امر مهم للاداة وهو الامر Start ويستخدم هذا الامر لبدء المؤقت ويستخدم الامر Stop لايقاف المؤقت .

تمرين :

قم بانشاء برنامج ساعة واضف اليه الادوات الاتية كما هو في الجدول :

الخاصية	القيمة	الشرح
Label1	Name	Lbltime
	Text	"العنصر"
Button1	Name	btnstart
	Text	"تشغيل"
Button2	Name	btnstop
	Text	"ايقاف"
Timer1	Interval	1000

• انظر الشكل ٤٠-٣ ولاحظ النموذج بعد اضافة الادوات وتغيير الخصائص فيها .



الشكل ٤٠-٣ برنامج الساعة

- اولاً : نقوم بكتابة كود التوقيت في اداة التوقيت في الحدث Tick() :

Visual Basic.Net **Code**

```
Public Class Form1

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Timer1.Tick
            If Timer1.Enabled = True Then
            lbltime.Text = "الساعة = " & TimeString

            Timer1.Start()

        Else

            Timer1.Stop()

        End If
    End Sub
End Class
```

ستلاحظ اننا قمنا باستخدام الجملة الشرطية وقمنا بالتحقق ما اذا كانت اداة التوقيت تم تمكينها ام لا ففي حال ثبت انها ممكنه هنا نفعل الامر بدء Start للتوقيت واذا كان غير ذلك بمعنى انها ليست ممكنه هنا نقوم بايقاف امر التوقيت من خلال الامر Stop .

- ثانيا : نقوم بكتابة كود تشغيل المؤقت في الزر btnstart

Visual Basic.Net **Code**

```
Public Class Form1

     Private Sub btnstart_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnstart.Click
        Timer1.Enabled = True
        btnstart.Enabled = False
        btnstop.Enabled = True
    End Sub
End Class
```

ففي الجملة الاولى من كود الزر btnstart قمنا بتمكين اداة التوقيت Timer ومنحناها القيمة True ثم في الجملة الثانية قمنا بالغاء تمكين الزر تشغيل وفي الجملة الثالثة قمنا بتمكين زر الايقاف .

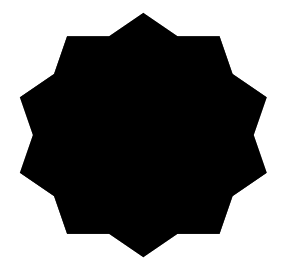
– ثالثاً : نقوم بكتابة كود ايقاف تشغيل المؤقت :

Visual Basic.Net **Code**

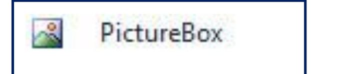
```
Public Class Form1

   Private Sub btnstop_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnstop.Click
        Timer1.Enabled = False
        btnstart.Enabled = True
        btnstop.Enabled = False
        lbltime.Text = "الساعة = " & "00:00:00"
    End Sub
End Class
```

ففي الجملة الاولى من كود الزر **btnstart** قمنا بالغاء تمكين اداة التوقيت **Timer** ومنحناها القيمة **False** ثم في الجملة الثانية بتمكين الزر تشغيل وفي الجملة الثالثة قمنا بالغاء تمكين زر الايقاف .



• اداة Picture Box



تستخدم اداةعرض الصور Picture Box في عرض الصور وتتيح للمبرمج التعامل مع الصور وتتيح له العديد من المزايا مثل امكانية التعديل واجراء الرسومات في الصور .

عزيزي القارئ هناك بعض الخصائص المتعلقة باداة عرض الصور Picture Box قد تم ذكرها في الادوات السابقة فلن نتطرق لذكرها مرة اخرى وهناك من الخصائص التي لم نذكرها سابقاً سنشرحها الان انظر الجدول الموجود اسفل :

الخاصية	القيمة	الشرح
– Backgroundimage		• تمكنك هذه الخاصية من تحديد الخلفية لاداة عرض الصور picture box وذلك من خلال استخدام الكود التالي : `Picturebox1.BackgroundImage = Image.FromFile(Application.StartupPath & "\vb.Net.jpg")`
– Backgroundimagelayout	– None – Tile – Center – Stretch – Zoom	• تمكنك هذه الخاصية من تحديد شكل ظهور الصورة داخل مربع الصورة Picturebox1. – عدم احداث شي . – تذييل – تحديد الخلفية بالوسط. – تاخذ الخلفية وضع امتدادي . – تكبير الخلفية .
– Image		• هذه الخاصية اشبه بالخاصية السابقة Backgroundimage الا انه توجد هناك فروق بسيطة .
– Size mode		• تعتبر هذه الخاصية اشبه بالخاصية السابقة Backgroundimagelayout الا

	– Normal – Stretch image – Auto Size – Center image – Zoom	أن هذي الخاصية متعلقة بالخاصية image وسنوضح لكم الفرق من خلال التمارين. – ظهور الصورة بالشكل العادي . – ظهور الصورة بشكل امتدادي . – التحكم التلقائي بشكل الظهور . – اظهار الصورة بالوسط . – تكبير الصورة .

سنوضح لكم اعزائي الخصائص السابقة للاداة Picture Box بمثال وستضح لكم الخصائص من خلال التمرين التالي

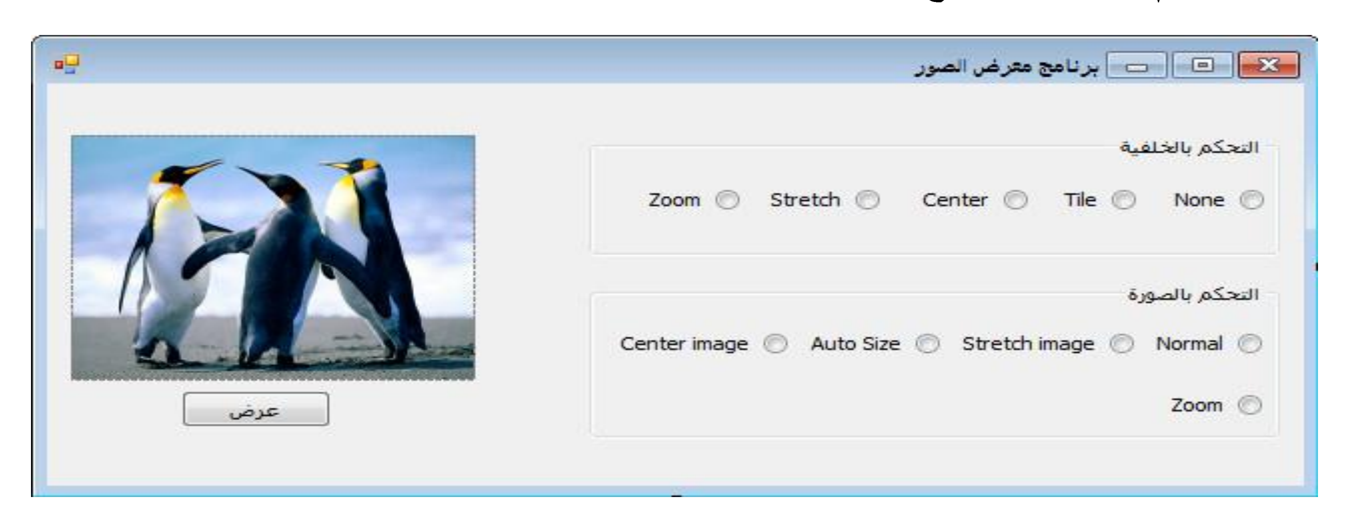
تمرين :

قم بانشاء برنامج يقوم بالتحكم بخصائص الاداة **Picture Box** السابقة وقم باضافة الادوات التالية بالجدول بالاسفل مع تغيير الخصائص المطلوبة للادوات :

الخاصية	القيمة	الشرح
Picture box1	Name	Pic1
GroupBox1	Text	"التحكم بالخلفية"
GroupBox2	Text	"التحكم بالصورة"
Button1	Name Text	Show "عرض"
Radio Button1	Name Text	Rd1 "None"
Radio Button2	Name Text	Rd2 "Tile"

Radio Button3	Name Text	Rd3 "center"
Radio Button4	Name Text	Rd4 "Stretch"
Radio Button5	Name Text	Rd5 "zoom"
Radio Button6	Name Text	Rd6 "Normal"
Radio Button7	Name Text	Rd7 "Stretch image"
Radio Button8	Name Text	Rd8 "Auto Size"
Radio Button9	Name Text	Rd9 "Center image"
Radio Button10	Name Text	Rd10 "zoom"

- اولاً : نقوم ببناء شكل البرنامج انظر الشكل ٤١-٣ :



الشكل ٤١-٣ يوضح شكل النموذج بعد اضافة الادوات .

- ثانياً : نقوم بتركيب الكود المتعلق بعرض الصورة show :

Visual Basic.Net **Code**

```
Public Class Form1


    Private Sub show_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
show.Click
        p1.Image = Image.FromFile(Application.StartupPath & "\m1.jpg")
    End Sub
End Class
```

- ثالثاً : نقوم بادخال الاكواد المتعلقة بالتحكم بالخلفية rd1,rd2,rd3,rd4,rd5 :

Visual Basic.Net **Code**

```
Public Class Form1

     Private Sub rd1_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles rd1.CheckedChanged
        p1.BackgroundImageLayout = ImageLayout.None

    End Sub

    Private Sub rd2_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles rd2.CheckedChanged
        p1.BackgroundImageLayout = ImageLayout.Tile
    End Sub

    Private Sub rd3_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles rd3.CheckedChanged
        p1.BackgroundImageLayout = ImageLayout.Center
    End Sub

    Private Sub rd4_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles rd4.CheckedChanged
        p1.BackgroundImageLayout = ImageLayout.Stretch
    End Sub

    Private Sub rd5_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles rd5.CheckedChanged
        p1.BackgroundImageLayout = ImageLayout.Zoom
    End Sub
End Class
```

- رابعاً : نقوم بادخال الاكواد المتعلقة بالتحكم بالصورة rd6,rd7,rd8,rd9,rd10 :

Visual Basic.Net	Code

```
Public Class Form1


    Private Sub rd6_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles rd6.CheckedChanged
        p1.SizeMode = PictureBoxSizeMode.Normal

    End Sub

    Private Sub rd7_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles rd7.CheckedChanged
        p1.SizeMode = PictureBoxSizeMode.StretchImage
    End Sub

    Private Sub rd8_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles rd8.CheckedChanged
        p1.SizeMode = PictureBoxSizeMode.AutoSize
    End Sub

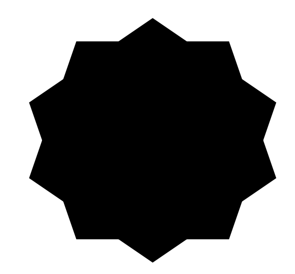
    Private Sub rd9_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles rd9.CheckedChanged
        p1.SizeMode = PictureBoxSizeMode.CenterImage

    End Sub

    Private Sub rd10_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles rd10.CheckedChanged
        p1.SizeMode = PictureBoxSizeMode.Zoom

    End Sub

End Class
```



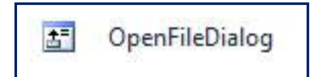
• اداة مربعات الحوار

تعتبر اداة مربعات الحوار اداة هامة يتم استخدامها في اعداد الخطوط ، والالوان ، وتصفح الملفات ، واستعراض المسارات ، وحفظ الملفات ، وفي طباعة المستندات وتظهر مربعات الحوار المتواجد معنا في بيئة **Visual Basic.Net** كالاتي :

- اداة الحوار لفتح الملفات
- اداة الحوار لتغيير الخط
- اداة الحوار لتغيير اللون
- اداة الحوار لحفظ الملفات
- اداة الحوار لاعداد الصفحة
- اداة الحوار للطباعة

سنقوم عزيزي القارئ بشرح الثلاث الحوارات الاولى وسنترك لك بقية الحوارات لتفهمها في مواضيع قادمة ان شاء الله .

• اداة حوار فتح الملفات



تستخدم هذه الاداة في استعراض وفتح الملفات بكافة انواعها للتعامل معها من خلال ادوات اخرى او لعرضها في ادوات اخرى مثلا أن نقوم بفتح ملفات الصور لعرضها في اداة **picture box** أو ملفات النصوص لعرضها في **text box** أو ملفات الميديا بلاير لعرضها في **windows media player** او ملفات الكتب الالكترونية أو ملفات الانترنت لعرضها في **web browser**

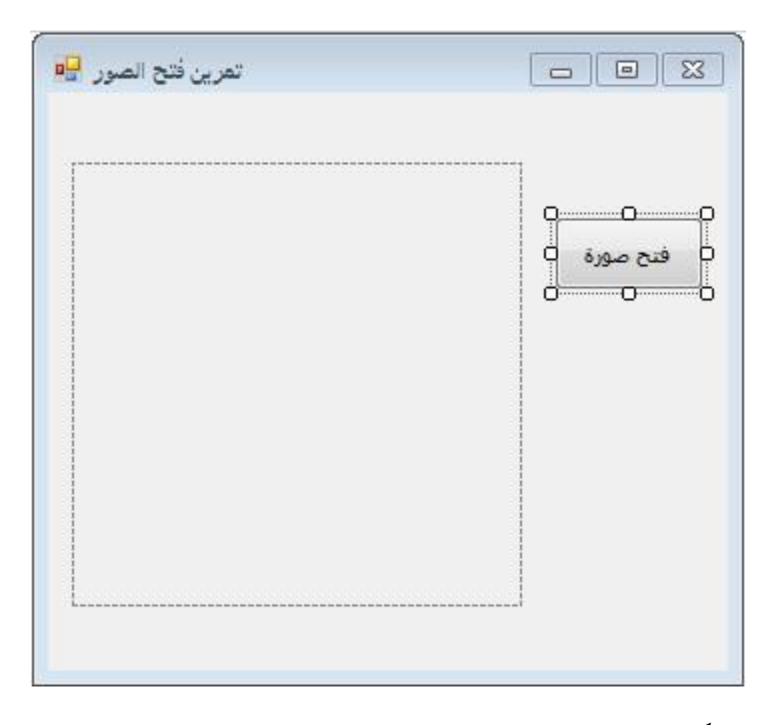
كل هذه الاستخدامات يمكن لنا أن نقوم بها باستخدام اداة فتح الملفات **open file dialog** سنوضح لكم حاليا مثال بسيط للتعامل مع اداة فتح الملفات :

<u>تمرين :</u>

قم بانشاء برنامج يقوم باستعراض الصور في الاداة **picture box** باستخدام اداة فتح الحوار **open file dialog** وقم باضافة الادوات التالية مع تغيير خصائصها حسب ماهو موجود في الجدول بالاسفل :

الخاصية	القيمة	الشرح
Picturebox1	Name	Pic1
Button1	Name	open
	Text	"فتح صورة"
OpenFileDialog	Name	ofd

- اولاً : نقوم ببناء شكل النموذج انظر الشكل ٤٢-٣ :



الشكل ٤٢ – ٣ يوضح برنامج فتح الصور

- ثانياً : بعد أن قمنا ببناء شكل النموذج نقوم الان بتركيب كود فتح الصور في الزر b1 :

Visual Basic.Net	Code

```
Public Class Form1

   Private Sub open_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
open.Click
        With ofd
            .CheckFileExists = True
            .ShowReadOnly = False
            .Filter = "All Files|*.*|Bitmap Files (*)|*.bmp;*.jpg"
            .FilterIndex = 1
            If .ShowDialog = DialogResult.OK Then
                pic1.Image = Image.FromFile(.FileName)
            End If
        End With
    End Sub
End Class
```

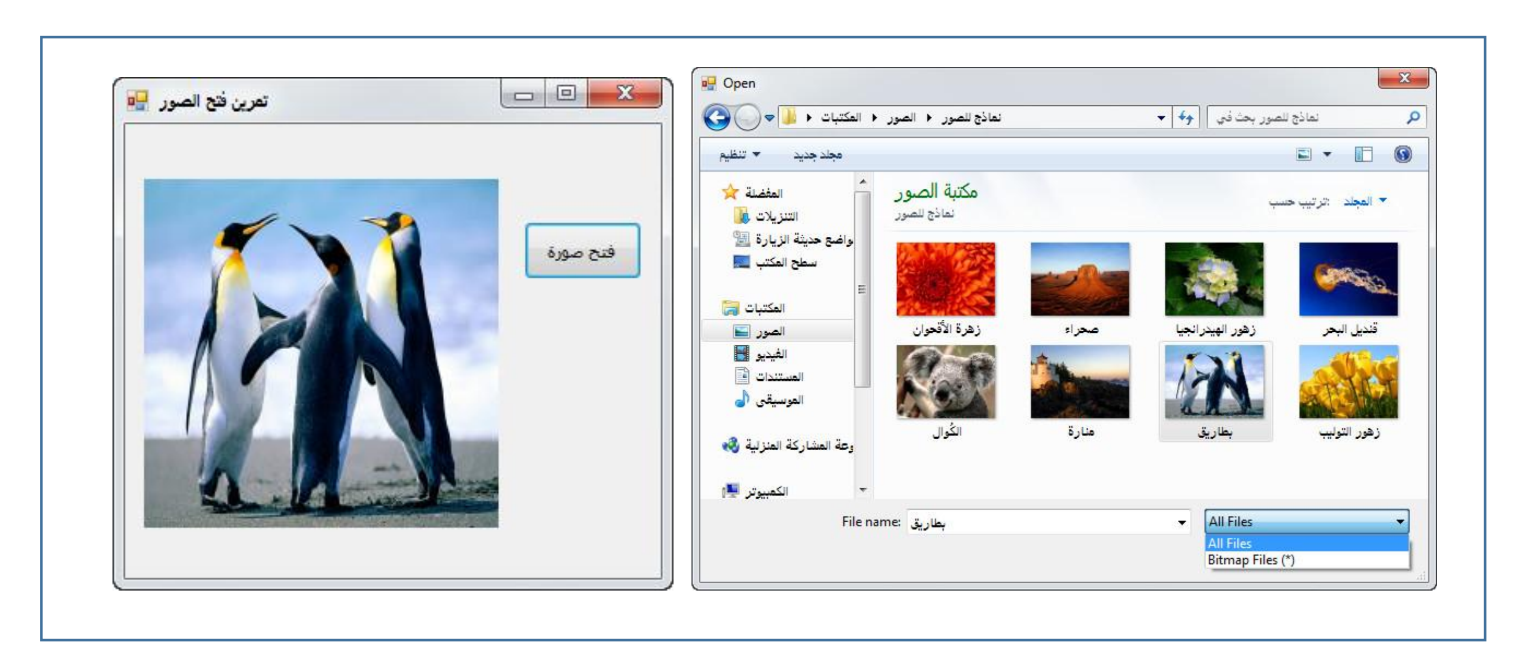
ملاحظة :

- استخدمنا في الكود السابق جملة with وختمناها بـ end with كي لانقوم بتكرار كتابة اسم اداة الحوار كل مرة .
- استخدمنا خاصية التاكد من الملف قبل فتحه .
- منعنا اظهار الملفات التي هي للقراءة فقط .
- حددنا نوع الملفات التي ستظهر في مربع الحوار وهي الملفات التي من نوع :

```
Bitmap Files (*)|*.bmp;*.jpg"
```

- بحال قام المستخدم بالنقر على الزر ok في مربع الحوار حينها سيقوم البرنامج بفتح الصورة التي اختارها .

انظر الشكل ٤٣ – ٣ ولاحظ شكل البرنامج عند فتح مربع الحوار وبعد استعراض الصورة :



الشكل ٤٣ – ٣ يوضح شكل البرنامج بعد فتح مربع الحوار واختيار الصورة وثم عرضها في picture box

- ## اداة حوار تغيير الخط



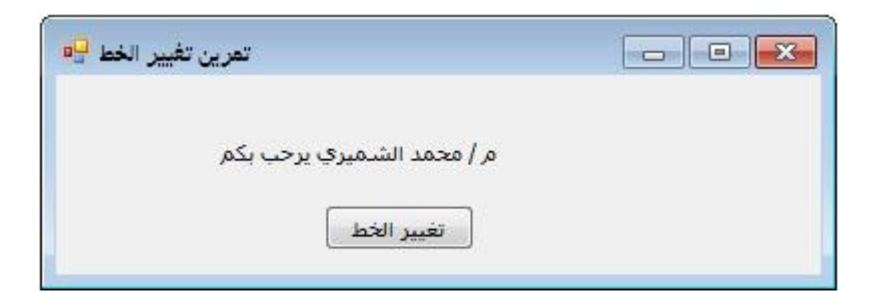
تعتبر اداة الحوار لتغيير الخط اسهل استخداماً وتعاملا من الاداة السابقة والفرق هو أن هذه الاداة تستخدم لتغيير خاصية الخطة لاداة محددة فاذا رغبت على سبيل المثال أن تقوم بتغيير خاصية الخط او الخط المتواجد في اداة **label** على سبيل المثال ماعليك الا أن تقوم باضافة اداة **label** وزر **button** واداة **font dialog**

تمرين :

قم بانشاء برنامج يقوم بتغيير نوع الخط المتواجد في اداة **label** وذلك باستخدام الاداة **font dialog** واضف اليه الادوات كما في الجدول بالاسفل :

الخاصية	القيمة	الشرح
Label1	Name Text	Label1 " م / محمد الشميري يرحب بكم"
Button1	Name Text	Button1 "تغيير الخط"
OpenFileDialog	Name	Font dialog1

- اولاً : نقوم ببناء شكل البرنامج انظر الشكل ٤٤ – ٣



الشكل ٤٤ – ٣ يوضح شكل البرنامج

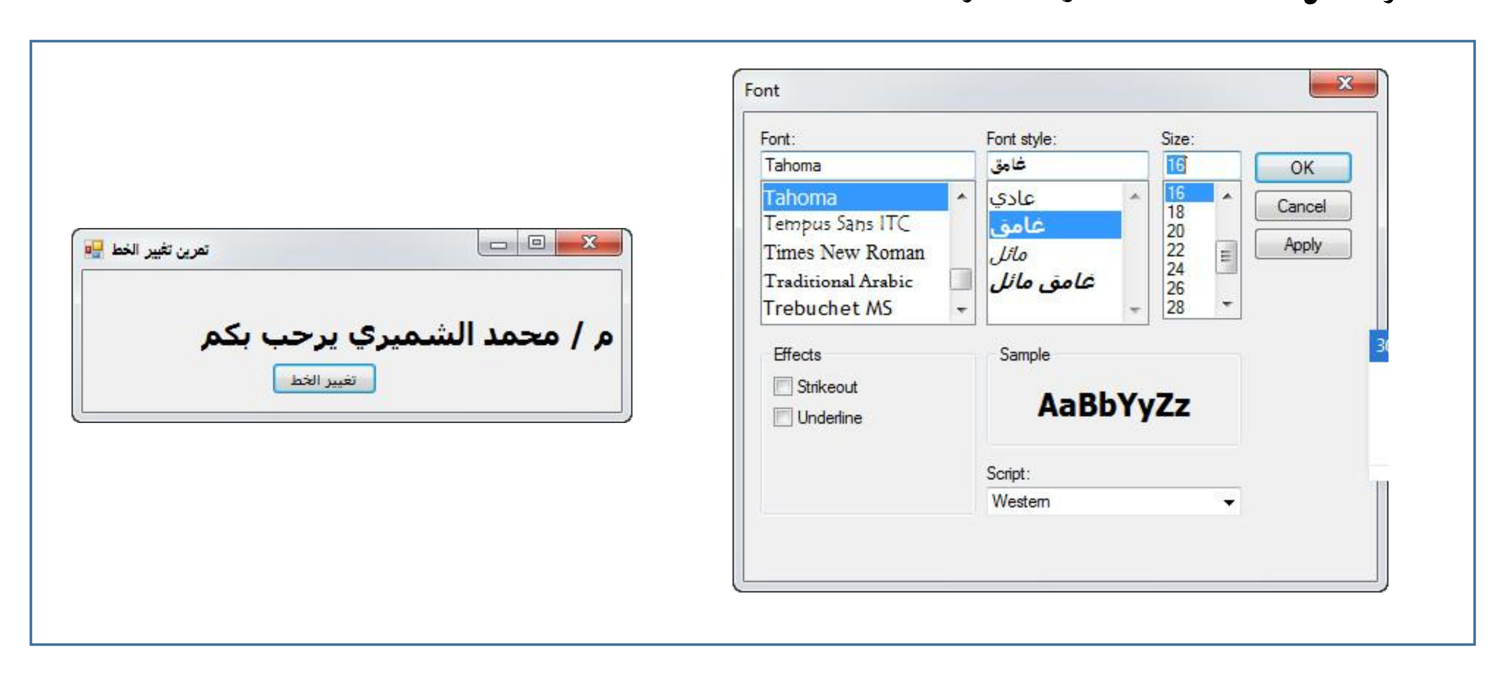
- ثانياً : نقوم بتركيب كود تغيير الخط في الزر **button1**

Visual Basic.Net	Code

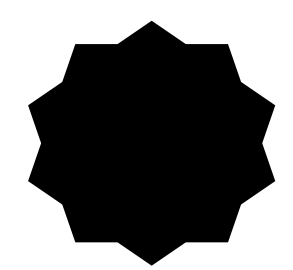
```
Public Class Form1

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
        With FontDialog1
            .ShowApply = True
            If .ShowDialog = DialogResult.OK Then
                Label1.Font = .Font
            End If
        End With
    End Sub
End Class
```

انظر الشكل ٤٥ – ٣ بعد التنفيذ وبعد اختيار الخط



الشكل ٤٥ – ٣ يوضح شكل البرنامج بعد تغيير خاصية الخط للنص الموجود في اداة **label1**





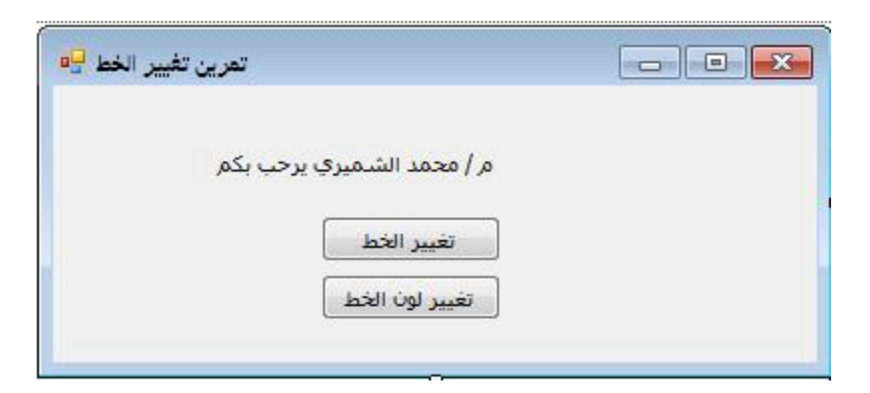
• اداة حوار تغيير اللون

لاتختلف اداة الحوار لتغيير اللون مع الاداة السابقة وتتشابه الى حد كبير معها الا ان هذه الاداة تستخدم لتغيير اللون فقط مثلا لون خلفية اداة معينة أو لون الخط .

تمرين :

من خلال الاستعانة بالتمرين السابق قم باضافة اداة تغيير اللون اليه واضف اليها زر **button** لتغيير لون الخط الموجود في **label**

- اولاً : قم نقوم باضافة الادوات المطلوب اضافتها للبرنامج السابق انظر الشكل ٤٦ – ٣ :



الشكل ٤٦ – ٣ يوضح شكل البرنامج بعد اضافة الزر **button2** لتغيير لون الخط

- ثانياً : نقوم بتركيب كود تغيير لون الخط في **button2**

Visual Basic.Net	Code

```
Public Class Form1

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button2.Click
        With ColorDialog1
            .SolidColorOnly = True
            If .ShowDialog = DialogResult.OK Then
                Label1.ForeColor = .Color
            End If
        End With
    End Sub
End Class
```



- ## اداة انشاء القوائم

ادا انشاء القوائم **menu strip** تستخدم هذه الاداة في انشاء قوائم تتضمن هذه القوائم مجموعة من الاوامر الي يتشابه استخدامها مثل اوامر الزر **button** الا أن هذه الاداة تقدم ميزة متعددة انها تمكن المستخدم من احتواء اكثر من امر واكثر من قائمة فرعية كما انها تستخدم لتمكين المستخدم بالتنقل مابين النماذج .

تمرين :

من خلال الاستعانة بالتمرين السابق قم باضافة اداة القائمة **menu strip** بحيث تحتوي القائمة على الاوامر التالية تغيير الخط والامر تغيير لون الخط .

- اولاً : نقوم باضافة اداة القائمة ونضيف اليها الاوامر المذكورة سابقا انظر الشكل ٤٧ – ٣



الشكل ٤٧ –٣ بعد اضافة القائمة والعناصر المطلوب اضافتها بالقائمة

- ثانياً : نقوم بتركيب الاكواد المتعلقة بتغيير الخط وبتغيير لون الخط في عناصر القائمة

Visual Basic.Net Code

```
Public Class Form1
Private Sub تغييرالخطToolStripMenuItem_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles تغييرالخطToolStripMenuItem.Click
        With FontDialog1
            .ShowApply = True
            If .ShowDialog = DialogResult.OK Then

                Label1.Font = .Font
            End If
        End With
    End Sub
    Private Sub تغييرلونالخطToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles تغييرلونالخطToolStripMenuItem.Click
        With ColorDialog1
            .SolidColorOnly = True
            If .ShowDialog = DialogResult.OK Then
                Button1.BackColor = .Color
                Label1.ForeColor = .Color
            End If
        End With
    End Sub
End Class
```

التعامل مع الاحداث Events

سنوضح لكم الان كيف تتعامل مع الاحداث Events بعد أن قمنا ببداية الفصل بايضاح نبذة مختصرة عن الاحداث فحالياً سنوضح لكم وبشكل اكبر التعامل مع الاحداث ولكل حدث غرض محدد الاستخدام تتشابه بعضها في بعض الادوات وتختلف في بعض الادوات الاخرى فعلى سبيل المثال اذا اخذنا اداة الزر button وقمنا بدراسة الاحداث المتوفرة لهذه الاداة سنسرد لكم بعض من احداث الاداة button وعن كيفية التعامل مع احداثها فعلا سبيل المثال الحدث click ويستخدم هذا الحدث عند النقر لمرة واحدة على الاداة ليقوم بعدها الزر بتنفيذ الامر الذي بداخله ، كذلك الز Double click ومعناها النقر المزدوج نستخدم هذا الحدث في حال رغبنا تنفيذ الامر المتواجد داخل الزر عند النقر مرتين بالزر الايسر للفارة على اداة button سنوضح لكم بالجدول الموجود بالاسفل عن تلك الاحداث والغرض منها :

الحدث	الوظيفة
– Click	• الحدث click وظيفته تنفيذ كود محدد عند القيام بعملية النقر على Button1
– Double click	• الحدث double click يقوم بتنفيذ كود محدد عند القيام بعملية نقر مزدوج على Button1 .
– Mouse Hover	• الحدث Mouse Hover وظيفته تنفيذ امر محدد عند المرور بمؤشر الماوس فوق الزر button .
– Mouse Leave	• الحدث Mouse Leave وظيفته تنفيذ امر محدد عند خروج مؤشر الماوس من فوق الزر button .

تمرين :

قم بانشاء برنامج له القدرة بالتعامل مع الاحداث المذكورة بالجدول السابق وذلك من خلال الاداةButton1 واعد تسمية الزر الى b1ومن ثم تعامل مع الاحداث المتعلقة به بتنفيذ الاوامر الموجودة بالجدول اسفل المتعلقة بالحدث والمهام المراد تنفيذها من الحدث:

الحدث	المطلوب تنفيذه من الحدث
– Click	قم بتغيير لون خلفية الزر B1 الى اللون الاحمر.
– Double Click	قم بتغيير لون خلفية الزر B1 الى اللون الاصفر.
– Mouse Hover	قم بتغيير لون خلفية الزر B1 الى اللون الازرق.
– Mouse Leave	قم بتغيير لون خلفية الزر B1 الى اللون الاخضر.

- اولاً : نقوم ببناء شكل البرنامج ونضف اليه الاداة **Button1** ونغير من خصائص اسمها الى **B1** كما هو مطلوب مننا انظر الشكل ٤٨–٣ :



الشكل ٤٨ – ٣ بعد اضافة الادوات اليه

- ثانياً : نقوم باستخراج الاحداث المطلوبة مننا كما هو بالجدول اعلاه وذلك من خلال النقر المزدوج على الاداة **B1** ثم نقوم باختيار الاحداث الخاصة بها لمراجعة شكل اختيار الاحداث راجع الشكل السابق ٩–٣ .

- ثالثاً : نقوم بتركيب اكواد الاحداث :

– الحدث **Click** :

Visual Basic.Net	Code

```
Public Class Form1


    Private Sub b1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
b1.Click
        b1.BackColor = Color.Red

    End Sub
End Class
```

– الحدث **Double Click** :

Visual Basic.Net	Code

```
Public Class Form1


     Private Sub b1_DoubleClick(ByVal sender As Object, ByVal e As System.EventArgs) Handles
b1.DoubleClick
        b1.BackColor = Color.Yellow
    End Sub
End Class
```

– الحدث Mouse Hover :

Visual Basic.Net **Code**

```
Public Class Form1



    Private Sub b1_MouseHover(ByVal sender As Object, ByVal e As System.EventArgs) Handles
b1.MouseHover
        b1.BackColor = Color.Blue

    End Sub
End Class
```

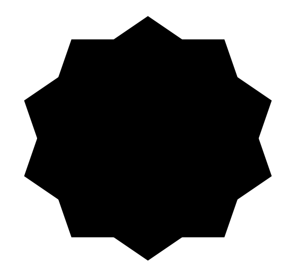
– الحدث Mouse Leave :

Visual Basic.Net **Code**

```
Public Class Form1


Private Sub b1_MouseLeave(ByVal sender As Object, ByVal e As System.EventArgs) Handles
b1.MouseLeave
        b1.BackColor = Color.Green

    End Sub
End Class
```



الفصل الرابع

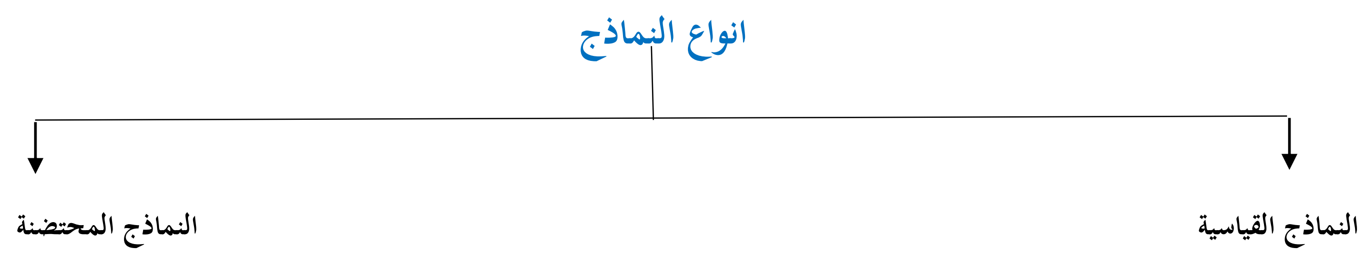
التعامل مع النماذج Forms

عزيزي القارئ سناخذك معنا بهذا الفصل الممتع الى برمجة ممتعة وهي برمجة النماذج أو كيفية التعامل مع خصائص هذه النماذج واحداثها واساليبها كما انه يجب عليك التركيز عزيزي القارئ على كل ماسنشرحه بهذا الفصل بالذات لانه يعتبر فصل بالنسبة للمبتدئيين، وانا متاكد جداً أن الجميع سيستمتع بهذا الفصل أكثر مني بكثير ، اعزائي سنتطرق بهذا الفصل الى :

- **مدخلك الى النماذج**
- **الخصائص والطرق والاحداث**
- **نماذج Midi Forms**
- **الثيمات (المظهر)**

مدخلك الى النماذج

النماذج هي عبارة عن شاشات لها اطار وهي جزء هام في المشاريع من النوع **windows application** المشاريع القياسية ويمكن من خلالها تحقيق معاالجات للمشكلات التي وجدت من اجلها وذلك من خلال مجموعة من الادوات والاكواد التي تضاف اليها .



Windows forms | **Midi forms**

– وسنتناول عزيزي القارئ كل نوع على حده بشرح اكثر تفصيل موضحين ذلك ببعض من الامثلة وسنبداء حاليا بشرح النوع الاول من النماذج **windows forms** :

النماذج القياسية Windows form

طريقة اضافة نماذج جديدة الى المشروع الحالي توجد طريقتين لاضافة النماذج الى المشروع :

- الطريقة الاولى : من خلال نافذة الحلول المتواجدة على يمين الشاشة انقر على المشروع الحالي فيها بالزر الايمن للفارة <- ثم اختار منها **Add** <- ثم اختار من القائمة الفرعية **Add item** اضافة عنصر <- ستظهر لك نافذة وهي نافذة اضافة العناصر وتختار منها **windows form** كما هو موضح لديك في (الاشكال من(١-٥)-٤)

- من خلال القائمة **project** اختار-> **add windows form** <- ثم اختار من نافذة اضافة العناصر **windows form**

خصائص النماذج windows form

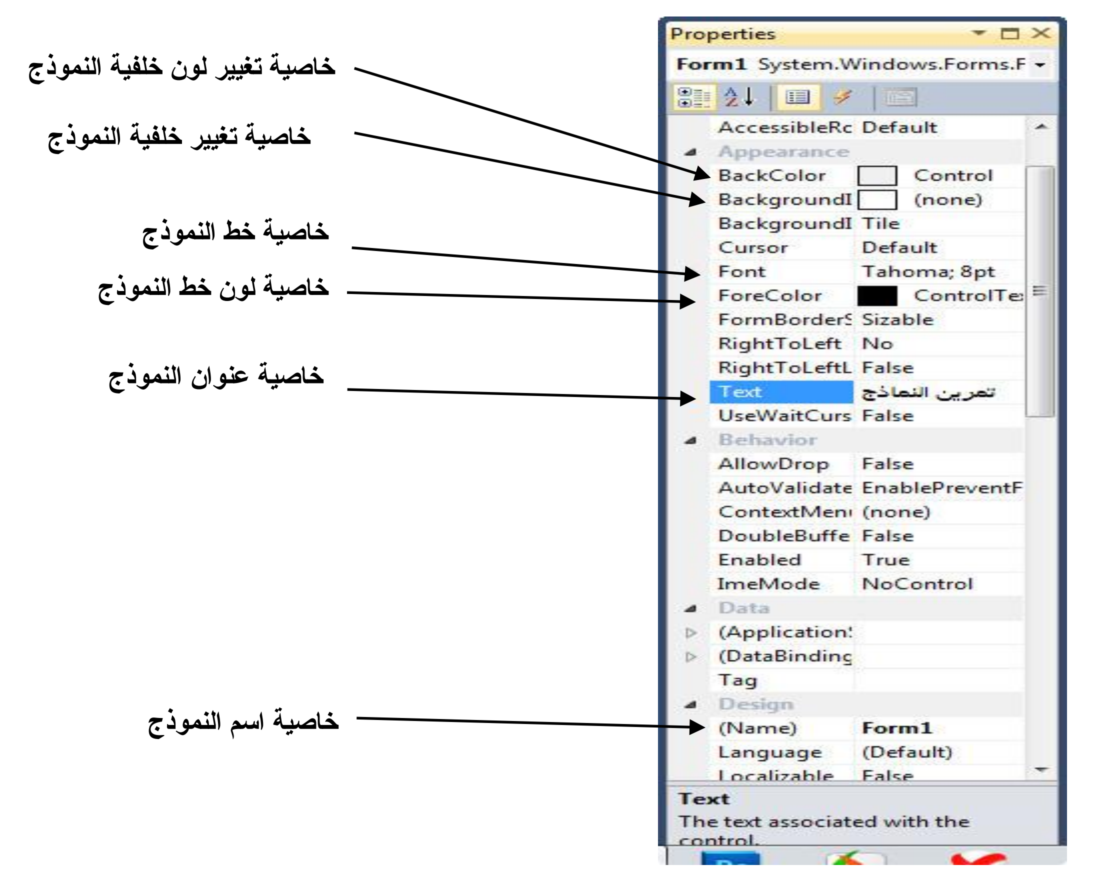
عزيزي القارئ تعد نافذة الخصائص نافذة مهمة جدا في بيئة التطوير Visual Basic.Net ، حيث وان هذه النافذة يمكن من خلالها كذلك التحكم بخصائص النماذج ، عزيزي القارئ لقد عرفت الكثير عن الخصائص في الفصل السابق وعن التعامل مع نافذة الخصائص وكذلك التعامل مع خصائص الادوات وكيفية تعيين الخصائص ؟ وطرق تعيين الخصائص ؟ لن يختلف هنا شئ بالنسبة للتعامل مع الخصائص عم سبق ولكن ستختلف بعض الخصائص فقط المتعلقة بالنماذج ونحن هنا في هذا الفصل سنوضح ذلك ،

طرق تعيين خصائص النماذج

كما اوضحنا في الفصل السابق لطرق تعيين الخصائص راجع طرق تعيين الخصائص بالفصل الاول ص ٨ وسنوضح لكم تعيين خصائص النماذج دون التطرق الى التفاصيل السابقة :

- الطريقة اليدوية :

وفي ظل هذه الطريقة سنوضح لكم الخصائص المتوفرة في نافذة الخصائص properties وتغيير القيم لتلك الخصائص انظر الشكل ١ – ٤ :



الشكل ١ – ٤ يوضح نافذة الخصائص وخصائص النموذج الموجودة فيها

ملاحظة :

عزيزي القارئ لن نتفرغ الى شرح تفاصيل هذه الطريقة ولكن بامكانك الرجوع الى كيفية تعيين الخصائص الطريق اليدوية بالفصل السابق .

- طريقة الاكواد :

تعتبر هذه الطريقة هي شبيه الى حد كبير بالطريقة ذاتها الموجودة بالفصل السابق لتعيين خصائص الادوات ولكن الفرق هنا هو اختلاف جزئي في بعض تلك الخصائص سنوضحها في الجدول الموجود بالاسفل :

الخاصية	القيمة	الشرح
– Form Border Style	– None – Fixed Single – Fixed 3D – Fixed Dialog – Sizable – Fixed Too Window – Sizable Too Window	• خاصية شكل الاطار : تمكنك هذه الخاصية من التحكم في شكل اطار النموذج . – عدم تغيير الشكل – منح النموذج شكل بسيط – منح النموذج شكل ثلاثي الابعاد . – منح النموذج شكل مربع الحوار مع ايقونة النموذج. – منح النموذج الشكل الطبيعي. – منح النموذج شكل النافذة مع اخفاء مقابس التكبير والتصغير – منح النموذج شكل النافذة مع ابقاء سمك الاطار بالحجم الطبيعي .
– Right to left–	– True – False	• خاصية اليمين الى اليسار : تعد هذه الخاصية هي اشبه بالخصاصية نفسها المتعلقة بالادوات الا أن هذه الخاصية تمكن من تحديد شكل ظهور الادوات الموجودة في النموذج – ظهور الادوات الموجودة في النموذج وعنوان النموذج من اليمين الى اليسار . – عدم ظهور الادوات الموجودة في النموذج وعنوان النموذج من اليمين الى اليسار

– Context menu strip		• تمكنك هذه الخاصية من ربط القوائم الفرعية الجانبية بالنموذج وسنشرح هذه الخاصية بشكل اكبر في تمارين قادمة .
– Start Position	– Manual – Center Screen – Windowsdefaultlocation – WindowsDefaultBounds – CenterParent	• موقع البدء : تمكنك هذه الخاصية من تحديد مكان ظهور النموذج في الشاشة. – ظهور النموذج بالجزء العلوي الايسر للشاشة . – ظهور النموذج في وسط الشاشة . – ظهور النموذج بالشكل العادي – ظهور النموذج بالشكل مكبر بالجزء العلوي الايسر – تتشابه مع الخاصية Windowsdefaultlocation
– Windowstate	– Normal – Minimized – Maximized	• حالة النافذة : تمكنك هذه الخاصية من تحديد حالة الظهور للنافذة . – حالة عادية . – حالة مصغرة. – حالة مكبرة.
– Control box	– True – False	• صندوق التحكم : تمكنك هذه الخاصية من التحكم في ظهور واخفاء مقابس الاطار الموجود في اطار النافذة العلوي والمحتوي على اغلاق ، وتكبير ، تصغير – ظهور صندوق التحكم – اخفاء صندوق التحكم .

– Icon		• الايقونة : تمكنك هذه الخاصية من تغيير ايقونة النموذج ico .
– Maximize box	– True – False	• صندوق التكبير : تمكنك هذه الخاصية من التحكم في تمكين خاصية تكبير النموذج بظهور او اخفاء صندوق التكبير الموجود في صندوق التحكم باعلى اطار النموذج . – ظهور صندوق التكبير . – اخفاء صندوق التكبير .
– Minimizebox	– True – False	• صندوق التصغير : تمكنك هذه الخاصية من التحكم في تمكين خاصية تصغير النموذج بظهور او اخفاء صندوق التصغير الموجود في صندوق التحكم باعلى اطار النموذج . – ظهور صندوق التصغير. – اخفاء صندوق التصغير.

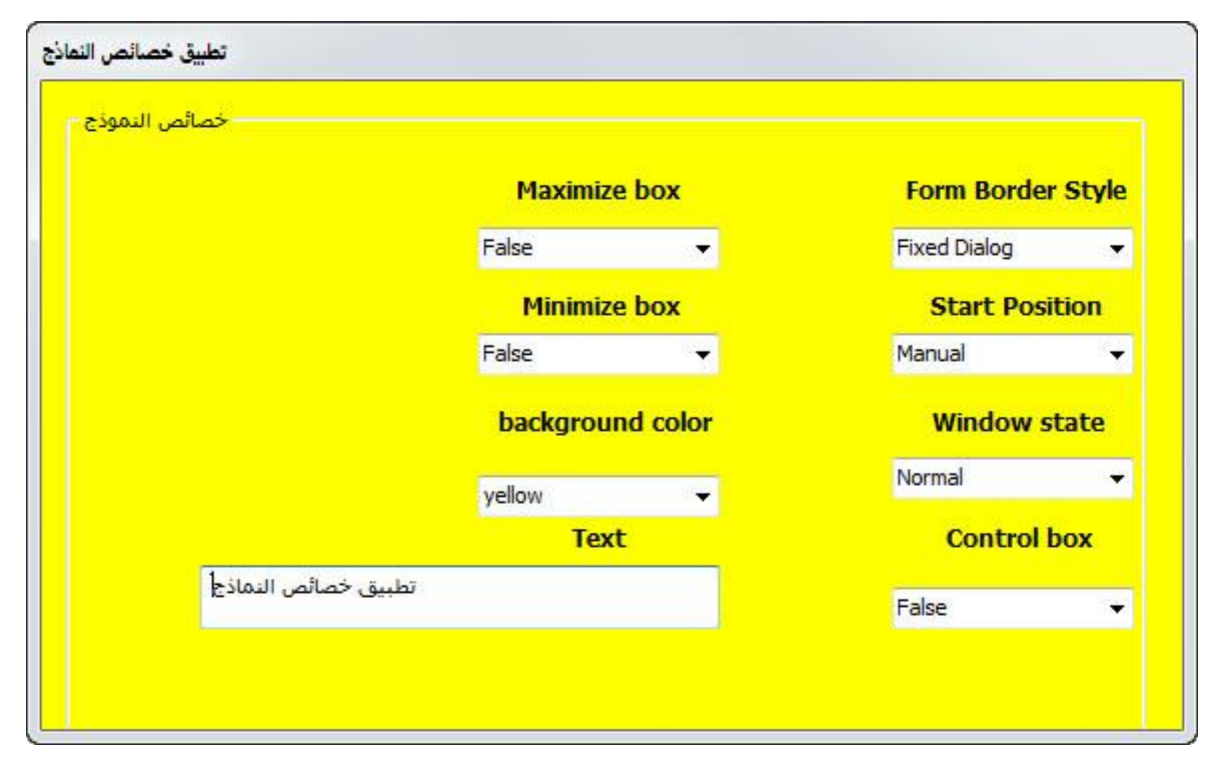
عزيزي القارئ سنقوم بايضاح لك الخصائص السابقة بالتمرين الاتي .

تمرين :

قم بانشاء برنامج يقوم بالتحكم بالخصائص التالية :

– Form Border Style ، Start Position ، Windowstate ، Control box، Maximize box ، Minimizebox ، **Text**,**backgroundcolor** وقم بتعيين قيم الخصائص لتلك :

الخاصية	القيمة	
ComboBox1	Items –	None – Fixed Single – Fixed 3D – Fixed Dialog – Sizable – Fixed Too Window – Sizable Too Window –
ComboBox2	Items –	Manual – Center Screen – Windowsdefaultlocation – WindowsDefaultBounds – Center Parent –
ComboBox3	Items –	Normal – Minimized – Maximized –
ComboBox4	Items –	True – False –
ComboBox5	Items –	True – False –
ComboBox6	Items –	True – False –
Textbox1	Text –	Text –
Label1	Text –	Form Border Style –
Label2	Text –	Start Position –
Label3	Text –	Window state –
Label4	Text –	Control box –
Label5	Text –	Maximize box –
Label6	Text –	Minimize box –
Label7	Text –	Text –



الشكل ٢-٤ يوضح كيف التحكم بالخصائص من خلال الكود

اليكم الكود البرمجي الخاص بالبرنامج :

Visual Basic.Net **Code**

```
Public Class Form1

    Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ComboBox1.SelectedIndexChanged
        On Error Resume Next
        Select Case ComboBox1.SelectedIndex
            Case 0
                Me.FormBorderStyle = Windows.Forms.FormBorderStyle.None
            Case 1
                Me.FormBorderStyle = Windows.Forms.FormBorderStyle.FixedSingle
            Case 2
                Me.FormBorderStyle = Windows.Forms.FormBorderStyle.Fixed3D
            Case 3
                Me.FormBorderStyle = Windows.Forms.FormBorderStyle.FixedDialog
            Case 4
                Me.FormBorderStyle = Windows.Forms.FormBorderStyle.Sizable
            Case 5
                Me.FormBorderStyle = Windows.Forms.FormBorderStyle.FixedToolWindow
            Case 6
                Me.FormBorderStyle = Windows.Forms.FormBorderStyle.SizableToolWindow
        End Select
    End Sub

    Private Sub ComboBox2_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ComboBox2.SelectedIndexChanged
```

```
        On Error Resume Next

        Select Case ComboBox2.SelectedIndex
            Case 0
                Me.StartPosition = FormStartPosition.Manual
            Case (1)
                Me.StartPosition = FormStartPosition.CenterScreen
            Case 2
                Me.StartPosition = FormStartPosition.WindowsDefaultLocation
            Case 3
                Me.StartPosition = FormStartPosition.WindowsDefaultBounds
            Case 4
                Me.StartPosition = FormStartPosition.CenterParent
                    End Select
    End Sub

    Private Sub ComboBox4_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ComboBox4.SelectedIndexChanged
        On Error Resume Next

        Select Case ComboBox4.SelectedIndex
            Case 0
                Me.ControlBox = True

            Case (1)
                Me.ControlBox = False

        End Select

    End Sub

    Private Sub ComboBox3_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ComboBox3.SelectedIndexChanged
        On Error Resume Next

        Select Case ComboBox3.SelectedIndex
            Case 0
                Me.WindowState = FormWindowState.Normal

            Case (1)
                Me.WindowState = FormWindowState.Maximized

            Case 2
                Me.WindowState = FormWindowState.Minimized

            Case 3

        End Select

    End Sub

    Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles TextBox1.TextChanged
        Me.Text = TextBox1.Text

    End Sub

    Private Sub ComboBox7_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ComboBox7.SelectedIndexChanged
        On Error Resume Next

        Select Case ComboBox7.SelectedIndex
            Case 0
```

```
                Me.BackColor = Color.Blue


            Case (1)
                Me.BackColor = Color.Yellow


        End Select
    End Sub

    Private Sub ComboBox5_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ComboBox5.SelectedIndexChanged
        On Error Resume Next

        Select Case ComboBox5.SelectedIndex
            Case 0
                Me.MaximizeBox = True

            Case (1)
                Me.MaximizeBox = False

        End Select
    End Sub

    Private Sub ComboBox6_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ComboBox6.SelectedIndexChanged
        On Error Resume Next

        Select Case ComboBox6.SelectedIndex
            Case 0
                Me.MinimizeBox = True

            Case (1)
                Me.MinimizeBox = False

        End Select
    End Sub

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load

    End Sub
End Class
```

ملاحظة :

عزيزي القارئ كما تلاحظ في الكود السابق تم استخدام الجملة **on error resume next** قبل كتابة كل كود برمجي من اجل تجنب الاخطاء البرمجية .

كذلك كما تلاحظون قمنا بتغيير خصائص النموذج من خلال وضع تلك الخصائص على شكل عناصر في القوائم المنسدلة Combo box واستخدمنا خاصية Selected index للتحقق من العناصر في القائمة المنسدلة.

الاحداث Events

نحن تحدثنا في الفصول السابقة عن الاحداث Events في الادوات وماذا تعني كلمة حدث ؟ عليك مراجعة ذلك في الفصل الثالث ص....

الان سنتعلم في هذا الفصل ماهي الاحداث المتعلقة بالنماذج ؟ وهل تتفق احداث النماذج مع احداث الادوات ؟

كذلك سنتعلم كيف نتعامل مع احداث النماذج ، بالامثلة والتطبيقات .

احداث النماذج :

الحدث	الشرح
Load	هو عبارة عن حدث تحميل النموذج ويتم من خلاله تنفيذ الاوامر أو الاكواد اثناء الوهلة الاولى من فتح البرنامج على سبيل المثال : • نقوم بانشاء برنامج رسالة ترحيب مبسط يقوم بعرض رسالة ترحيبية اثناء فتح البرنامج . ماعلينا الا أن نقوم بالنقر المزدوج على صندوق التحكم باعلى النموذج لكي يعرض لنا الحدث Load ومن ثم نكتب كود الرسالة الترحيبية : **Msgbox**("**welcome**")
form closed	هو عبارة عن حدث ينفذ امر ما بحال رغبة المستخدم أن ينفذ هذا الامر اثناء خروجه من النموذج .
form closing	يتشابه هذا الحدث بشكل كبير مع الحدث السابق form closed
click	يستخدم هذا الحدث بحال رغبة المستخدم تنفيذ امر معين عند النقر باي مكان على النموذج .
Mouse hover	يستخدم هذا الحدث بحال رغبة المستخدم تنفيذ امر معين عند مرور مؤشر الفارة من فوق النموذج .

تتشابه الى حد كبير بعض من احداث النموذج مع احداث الادوات وهو كما لاحظتم من خلال الجدول السابق ونحن اوجزنا الاحداث لنترك للقارئ البحث والاطلاع والممارسة على بقية الاحداث وسنوضح لك عزيزي القارئ الان مثال بسيط عن ذلك :

تمرين :

قم بانشاء برنامج يظهر رسالة ترحيبية اثناء الدخول للبرنامج وبنفس الوقت يظهر رسالة تاكيدية بان المستخدم يرغب بالخروج من البرنامج وفي حال تاكد بانه يرغب بالخروج يتم تنفيذ امر الخروج .





الشكل ٣-٤ يوضح ظهور الرسالة الترحيبية عند فتح البرنامج

اليكم الكود البرمجي الخاص به

Visual Basic.Net **Code**

```
Public Class Form1

    Private Sub Form1_FormClosed(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosedEventArgs) Handles Me.FormClosed
    If MsgBox("هل تريد الخروج من البرنامج", MsgBoxStyle.OkCancel) = MsgBoxResult.Cancel Then
            Exit Sub
        Else
            End

        End If


    End Sub

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        MsgBox("مرحبا بك في برنامجنا")
    End Sub
End Class
```

الاساليب Methods

الاساليب Methods هي عبارة عن الافعال التي يستطيع اجرائها أو هي الافعال التي يمكن على العناصر وتشترك عناصر التحكم بمجموعة من الاساليب وتختلف بمجموعة اخرى ومن الاساليب التي تشترك فيها :

- **Show , hide , Refresh**

الاسلوب	وظيفته
Show	يقوم هذا الاسلوب بـ عرض النماذج ولكن عدم ابقائها في المقدمة .
hide	يقوم هذا الاسلوب بـ اخفاء النماذج .
Refresh	يقوم هذا الاسلوب بـ تحديث النماذج .

ومن اكثر الاساليب استخداماً :

- **Focus**
- **Refresh**

الصيغة العامة لاستدعاء الاساليب :

Object Name .Method



اسم العنصر اسم الاسلوب

فلا توجد علامة = على يمين الجملة لانه لايمكن تعيين القيمة للاسلوب .

مثال :

عندما نقوم باستدعاء وعرض احد النماذج نكتب هذه الجملة

Form1.show

أو

Me.hide

التنقل بين النماذج :-

تتيح لك لغة Visual Basic.Net امكانية التنقل بين النماذج من خلال الاوامر التالية :

- **Show ,hide ,Show dialog , End ,Close**

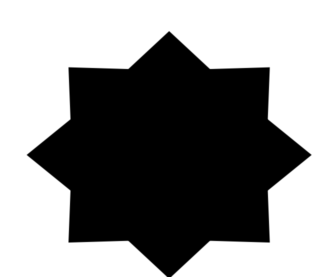
Show : تستخدم لعرض النموذج مع عدم الاحتفاظ بخاصية البقاء.

hide: تستخدم لاخفاء النموذج.

Show Dialog : تستخدم لعرض النموذج مع الاحتفاظ بخاصية البقاء.

End : تستخدم للخروج من البرنامج نهائيا.

Close: تستخدم للخروج من نموذج محدد.



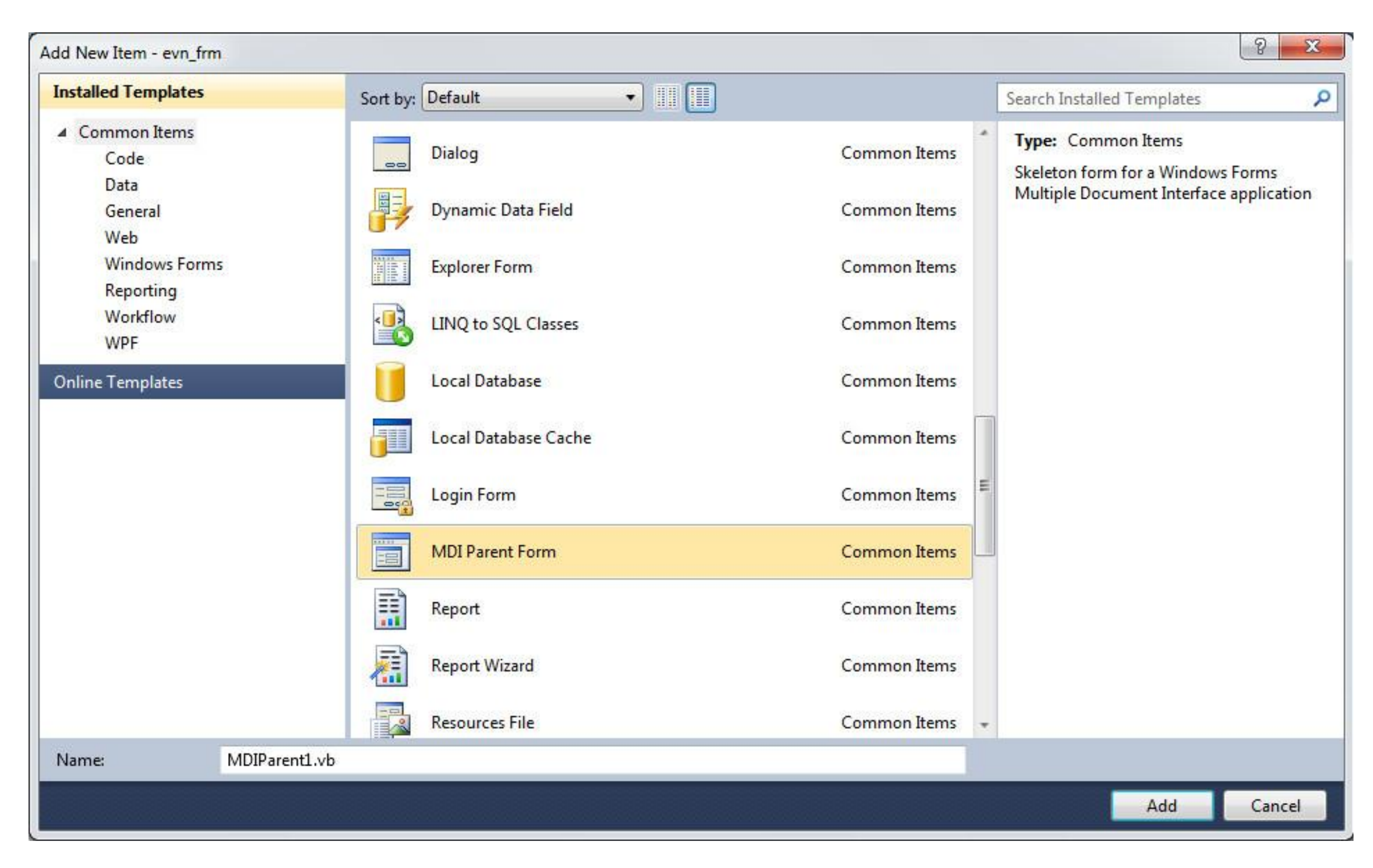
نماذج Midi Parent Form :-

لاتختلف كثيراً هذه الانواع من النماذج الحاظنة مع النماذج العادية سوا فوارق بسيطة حيث أن هذه النماذج تحتظن أي نماذج اخرى تقوم بفتحها بمعنى ان تلك النماذج لاتستطيع أن تسحبها الى خارج حدود النموذج الاب parent بعكس النماذج العادية التي تستطيع اجرى ذلك عليها وبامكانك سحب أي نموذج تقوم بفتحه الى خارج حدود النموذج الرئيسي .

اضافة نموذج Midi Parent form الى مشروعك :

لاتختلف طريقة اضافة هذا النوع من النماذج الى المشروع كـ النموذج العادي أو القياسي windows form اليكم الخطوات ذاتها :

من قائمة project <= نختار add windows form لتظهر لنا هذه النافذة كما بالشكل اسفل شكل ٤ – ٤ لنختار منها العنصر midi parent form .



شكل ٤ – ٤ يوضح طريقة اضافة نموذج midi parent form

الثيمات (المظهر)

هي عبارة عن اداة اضافية نقوم باستيرادها من خارج المشروع ووظيفة هذه الاداة تحسين مظهر المشروع ليكون اكثر جمالاً وتوجد ادوات كثيرة بهذا الجانب واغلب هذه الادوات تاتي متكاملة بمعنى انها تشمل لمعظم الادوات المتواجدة معنا في صندوق الادوات وسنوضح لكم مثال بسيط عن تلك الثيمات :

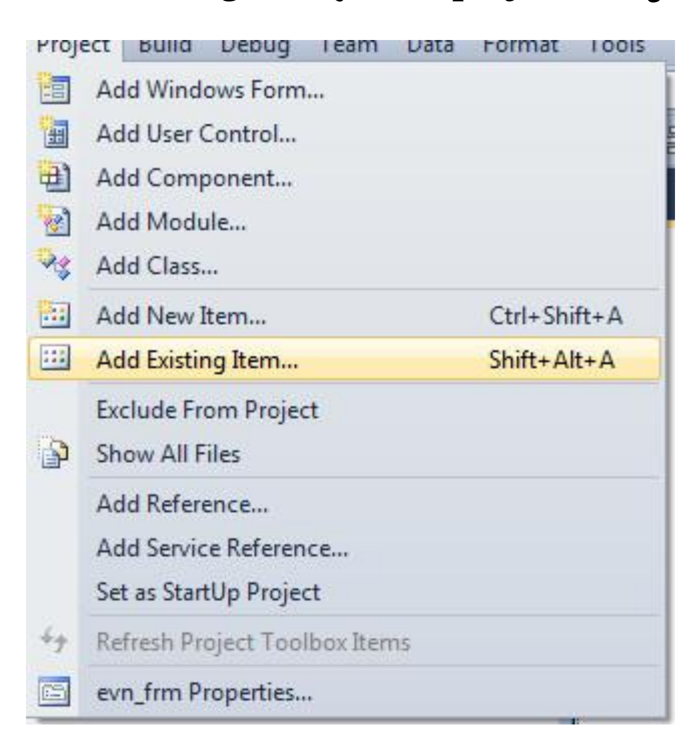
- facebook.vb
- Them Win8 BY Security.vb

هذه كـ مثال بسيط وتوجد العديد من هذه الثيمات .

تمرين :

قم باضافة برنامج واضف اليه الثيمات السابقة

من قائمة project => نختار add existing item كما بالشكل اسفل الشكل ٧ – ٤

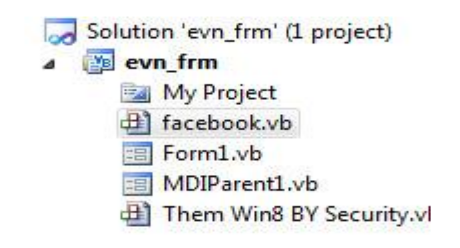


الشكل ٧ –٤ يوضح اختيار add existing item

ومنها نستورد الثيمات الى المشروع ونبحث عن مكان تواجدها ونحددها ومن ثم ننقر على **add** كما بالشكل ٨ – ٤

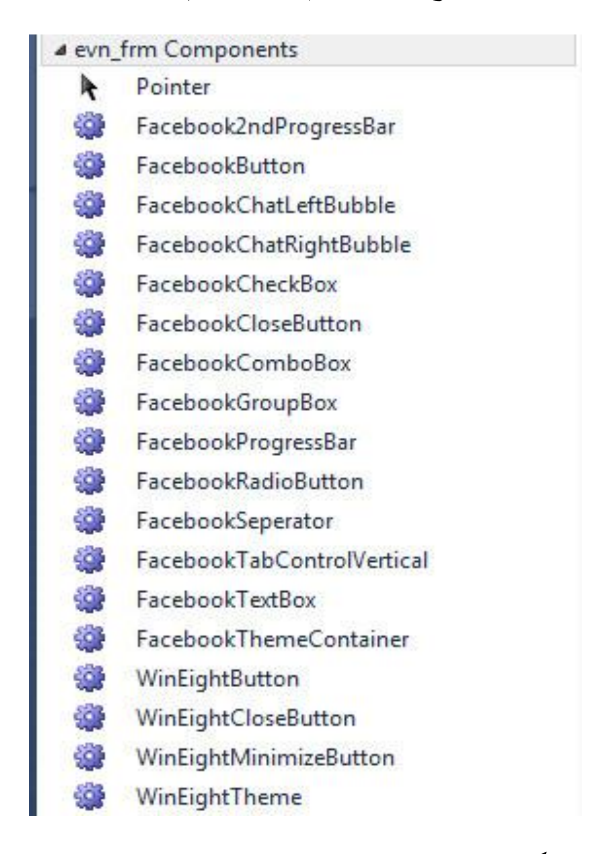
الشكل ٨– ٤ يوضح استيراد الثيمات الى المشروع

وبعد هذه الخطوة نقوم بالنقر المزدوج على عنصر الثيم المتواجد ضمن عناصر الحل للمشروع كم تلاحظوا بالشكل ٩ – ٤ بالاسفل من اجل اظهار الثيم ضمن صندوق الادوات.



الشكل ٩ –٤ يوضح طريقة اختيار عنصر الثيم

بعد أن قمنا بالنقر المزدوج على عنصر الثيم المراد اضافته الى صندوق الادوات نقوم بعدها بالضغط على الزر **F5** ليفتح البرنامج ونقوم باغلاق البرنامج مباشرة ليتم اضافة الثيم الى صندوق الادوات كما بالشكل ١٠ – ٤



الشكل ١٠ – ٤ يوضح اضافة عناصر الثيم الى صندوق الادوات

بعدها في حال رغبنا في تغيير شكل النموذج نقوم باضافة العنصر المتواجد بالشكل اعلاه الى النموذج **Face book theme container** نضيفها للنموذج ليصبح شكله كما في الشكل ١١ – ٤



الشكل ١١ – ٤ يوضح تغيير مظهر النموذج

لحماية الملف من التلف ولتجنب ظهور مشاكل تلف البرنامج اثناء التعامل مع هذا النوع من الادوات نقوم باقفال النموذج ومايحتويه من ادوات من خلال النقر بالزر الايمن للفارة ومن ثم اختيار **Lock controls** وبامكاننا اضافة ادوات الى النموذج ادوات مظهر الثيم المتواجد معنا من صندوق الادوات كما بالشكل ١٢ – ٤

Form1

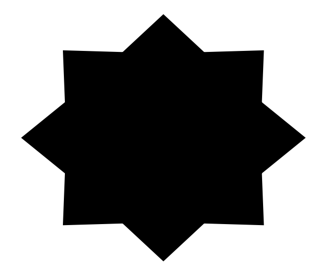
المستخدم

كلمة المرور

تسجيل الدخول

خروج

الشكل ١٢ – ٤ بعد اضافة ادوات الثيم اليه



المصادر :

- كتاب خطوة بخطوة مع Visual Studio 2008 ، احمد خليفة
- برمجة اطار عمل Net. باستخدام Visual Basic.Net، تركي العسيري


الحمد لله